

**UNIVERSIDAD CARLOS III DE MADRID**  
**ESCUELA POLITÉCNICA SUPERIOR**

**ITT: IMAGEN Y SONIDO**



**PROYECTO FIN DE CARRERA**

**VISUALIZACIÓN BIDIMENSIONAL DE PROBLEMAS DE  
CLASIFICACIÓN EN ALTA DIMENSIÓN**

**Autor: Javier Gijón Gómez**

**Tutor: Ángel Navia Vázquez**



## **Agradecimientos**

En primer lugar quisiera agradecer a mi Tutor Ángel Navia la oportunidad de desarrollar este proyecto y ayudarme a la consecución de mis estudios universitarios.

A todos mis compañeros con los que he compartido mi tiempo, mis prácticas, mis dudas, noches de fiesta, cenas y mis horas de estudio: Abel, Luis, Alberto C., Alberto S., Oscar, Pachi, Esther, María, Cristina, Aurora y Miriam. Y en especial a Víctor J. con el que he pasado grandes momentos dentro y fuera de la universidad.

A Víctor García Fernández, amigo de toda la vida, que me ha demostrado que la gente que de verdad te aprecia siempre estará a tu lado en lo bueno y en lo malo, cuando lo pidas y sobre todo cuando digas que quieres estar solo cuando en realidad quieres estar acompañado.

Agradecer a toda mi familia su comprensión, su atención cuando la necesite, los tirones de oreja cuando fueron merecidos y una palabra de apoyo en el momento oportuno. Gracias a todos ellos por hacer que lo que tuvieron que haber sido tres años sobre el papel se hayan llevado de la mejor de las formas.

A Ana que ha estado demasiado tiempo viéndome estudiar, quejarme, sofocarme, estresarme, maldecir y perder los nervios y aún así escucharme, comprender, quererme y por posponer nuestros planes de vida para facilitar la finalización de mis estudios.

Y finalmente hablar de una despedida y un encuentro, a mi gato Tenor que me dejó tras 16 años de inestimable compañía y no ha podido ver la consecución de tantos años de estudio. Y la bienvenida a Apolo, un perro de 4 meses al que se le ha dado una nueva oportunidad en la vida.

A todos gracias por cada grano de arena que habéis aportado en mi vida durante este tiempo.

## **Resumen**

El objetivo de este proyecto es obtener buenas representaciones en dos dimensiones de problemas  $N$  dimensionales. Para ello se propone una función de coste que mida la similitud entre una representación en  $N$  dimensiones y la misma en 2 dimensiones. Esto permite que se pueda realizar una comparación de la eficacia y una clasificación de diferentes técnicas de reducción de dimensionalidad para descubrir en distintos conjuntos cuales de estas son las que mantienen un mayor grado de similitud entre ambos espacios dimensionales.

**Palabras clave:** Visualización, 2D, bidimensional, alta dimensión, reducción de dimensionalidad.

## **Abstract**

The main objective of this Project is to get proper good-quality representations in two dimensions of N dimensional problems. In order to do that, a cost function is proposed to measure the similarity between a representation in N dimensions and the same in two dimensions. ; that allows us to be able to carry out an effectiveness comparison and a classification of different techniques to reduce 'dimensionality' so as to find out, in different data sets, which of those are the ones that keep an upper similarity level between both dimensional spaces.

**Keywords:** Visualizing, 2D, bldimensional, high-dimension, dimensionality reduction.

## Índice de tablas

Tabla 2.1. Trabajos que han tenido éxito usando LIBSVM.....	15
Tabla 2.2. Propiedades de las técnicas de reducción de dimensionalidad .....	20
Tabla 2.3. Generalización de errores de clasificadores de vecinos cercanos en representaciones de baja dimensionalidad de la base de datos MNIST, la base de datos characters y de la base de datos 20 newsgroups.....	29
Tabla 3.1. Valores obtenidos de los parámetros libres en varias ejecuciones del ejemplo wine.....	39
Tabla 3.2. Técnicas de reducción de dimensionalidad 1 de 3 .....	40
Tabla 3.3. Técnicas de reducción de dimensionalidad 2 de 3 .....	41
Tabla 3.4 Técnicas de reducción de dimensionalidad 3 de 3 .....	42
Tabla 3.5. Colores en matlab .....	44
Tabla 3.6. Ejemplo de Kendall-Tau .....	56
Tabla 3.7. Esquema de la realización de asignación de puntos en Kendall-Tau .....	56
Tabla 3.8. Métodos para los que no se han obtenido resultados en iris .....	60
Tabla 3.9. Resultados para el conjunto de datos iris 1 de 3 .....	60
Tabla 3.10. Resultados para el conjunto de datos iris 2 de 3 .....	61
Tabla 3.11. Resultados para el conjunto de datos iris 3 de 3 .....	62
Tabla 3.12. Métodos para los que no se han obtenido resultados en wine .....	62
Tabla 3.13. Resultado para el conjunto de datos wine 1 de 3 .....	63
Tabla 3.14. Resultados para el conjunto de datos wine 2 de 3 .....	64
Tabla 3.15. Resultados para el conjunto de datos wine 3 de 3 .....	65
Tabla 3.16. Métodos para los que no se han obtenido resultados en car.....	65
Tabla 3.17. Resultado para el conjunto de datos car 1 de 3.....	66
Tabla 3.18 Resultados para el conjunto de datos car 2 de 3.....	67
Tabla 3.19. Resultados para el conjunto de datos car 3 de 3.....	68
Tabla 3.20. Métodos para los que no se han obtenido resultados en balance .....	68
Tabla 3.21. Resultado para el conjunto de datos balance 1 de 3 .....	69
Tabla 3.22. Resultados para el conjunto de datos balance 2 de 3 .....	70
Tabla 3.23. Resultados para el conjunto de datos balance 3 de 3 .....	71
Tabla 3.24. Resultados ordenados de mayor a menor eficiencia para cada conjunto de datos 1 de 2.....	72
Tabla 3.25. Resultados ordenados de mayor a menor eficiencia para cada conjunto de datos 2 de 2.....	73

# Índice de Figuras

Figura 2.1: Problemas primigenios en la clasificación de vectores .....	16
Figura 2.2: Formulas del problema dual en la clasificación de vectores .....	17
Figura 2.3. Condición para la obtención de una $w$ óptima .....	17
Figura 2.4 Condición para que el problema dual sea realizable en V-SVC .....	17
Figura 2.5 Formulas de las funciones de decisión para clasificación en SVM .....	18
Figura 2.6 Versión escalada del problema dual .....	18
Figura 2.7 Gráfico de las técnicas de reducción de dimensionalidad .....	19
Figura 2.8. Función objetivo de técnicas convexas .....	20
Figura 2.9. Función de minimización de LTSA .....	25
Figura 2.10 Versión simple de la distribución t-SNE .....	28
Figura 2.11. Visualización de 10.000 dígitos de la base de datos MNIST con técnicas paramétricas de reducción de dimensionalidad .....	28
Figura 2.12. Visualización de la base de datos de las caras de Olivetti .....	29
Figura 2.13. Visualización de la base de datos COIL-20 .....	29
Figura 3.1 Captura de pantalla modifica del repositorio de datos .....	35
Figura 3.2. Imagen de ejemplo para un conjunto con tasa de acierto baja .....	46
Figura 3.3. Ejemplo car para una iteración mediante t-SNE .....	50
Figura 3.4. Ejemplo iris para una iteración mediante LDA .....	50
Figura 3.5. Función del cálculo de Kendall-Tau .....	55
Figura 3.6. Función y resultado del ejemplo de Kendall-Tau propuesto .....	57
Figura 3.7. Imagen del conjunto iris sobre el método de SNE .....	75
Figura 3.8. Imagen del conjunto iris sobre el método de LDA .....	75
Figura 3.9. Imagen del conjunto iris sobre el método de tSNE .....	76
Figura 3.10. Imagen del conjunto iris sobre el método de Diff. Maps .....	76
Figura 3.11. Imagen del conjunto iris sobre el método de SymSNE .....	77
Figura 3.12. Imagen del conjunto wine sobre el método de tSNE .....	77
Figura 3.13. Imagen del conjunto wine sobre el método de MCML .....	78
Figura 3.14. Imagen del conjunto wine sobre el método de LPP .....	78
Figura 3.15. Imagen del conjunto wine sobre el método de NPE .....	79
Figura 3.16. Imagen del conjunto wine sobre el método de SymSNE .....	79
Figura 3.17. Imagen del conjunto balance sobre el método de tSNE .....	80
Figura 3.18. Imagen del conjunto balance sobre el método de MCML .....	80
Figura 3.19. Imagen del conjunto balance sobre el método de Laplacian .....	81
Figura 3.20. Imagen del conjunto balance sobre el método de NPE .....	81

Figura 3.21. Imagen del conjunto balance sobre el método de SymSNE ..... 82

Figura 3.22. Imagen del conjunto car sobre el método de NCA ..... 82

Figura 3.23. Imagen del conjunto car sobre el método de KernelPCA ..... 83

Figura 3.24. Imagen del conjunto car sobre el método de LDA..... 83

Figura 3.25. Imagen del conjunto car sobre el método de ProbPCA..... 84

Figura 3.26. Imagen del conjunto car sobre el método de PCA ..... 84



# Índice

Capítulo 1: Introducción .....	11
1.1 Motivación del proyecto.....	11
1.2 Objetivos .....	12
1.3 Contenido de la memoria .....	13
Capítulo 2: Estado del arte.....	14
2.1 Introducción.....	14
2.2 LIBSVM .....	14
2.2.1 Formulaciones de SVM.....	15
2.3 Técnicas de reducción de dimensionalidad .....	19
2.3.1 Técnicas de reducción de dimensionalidad convexas .....	20
2.3.2 Técnicas espectrales dispersas .....	23
2.3.3 Técnicas de reducción de dimensionalidad no convexas .....	25
Capítulo 3. Desarrollo del proyecto .....	31
3.1 Introducción.....	31
3.2 Primeros pasos .....	31
3.3 Aumento en la complejidad de la implementación .....	36
3.5 Descripción de los conjuntos de datos .....	43
3.5.1 Conjunto Iris.....	47
3.5.2 Conjunto Wine .....	47
3.5.3 Conjunto balance .....	47
3.5.4 Conjunto Car.....	48
3.6 Primeras visualizaciones de datos.....	49
3.7 Método de medición entre modelos dimensionales .....	51
3.7.1 Estado inicial.....	52
3.7.2 Reducción a dos dimensiones .....	53
3.7.3 Tasa de acierto .....	54
3.7.4 Kendall- $\tau$ .....	55
3.7.5 Comparación de los errores.....	57
3.7.6 Valor de salida .....	58
3.8 Resultados obtenidos .....	59

3.8.1 Ejemplo iris .....	60
3.8.2 Ejemplo wine .....	62
3.8.3 Ejemplo car .....	65
3.8.4 Ejemplo balance .....	68
3.8.5 Resultados de salida clasificados .....	72
3.9 Representación gráfica de resultados .....	73
3.9.1 Representaciones del conjunto Iris .....	75
3.9.2 Representaciones del conjunto Wine .....	77
3.9.3 Representaciones del conjunto Balance .....	80
3.9.4 Representaciones del conjunto Car .....	82
Capítulo 4: Conclusiones y trabajos futuros .....	85
4.1 Conclusiones.....	85
4.2 Trabajos futuros .....	86
Capítulo 5: Anexos .....	88
5.1 Representaciones .....	88
5.1.1 Conjunto iris.....	88
5.1.2 Conjunto Wine .....	96
5.1.3 Conjunto Balance.....	104
5.1.4 Conjunto Car.....	111
5.2 Resultados numéricos ampliados.....	118
5.2.1 Conjunto de datos Iris .....	119
5.2.2 Conjunto de datos Wine.....	134
5.2.3 Conjunto de datos Balance .....	151
5.2.4 Conjunto de datos Car .....	165
Capítulo 6: Bibliografía.....	178

# Capítulo 1: Introducción

## 1.1 Motivación del proyecto

La realización de este proyecto viene motivada por diversas causas. La primera de ellas es que muchos problemas de clasificación vienen determinados por un gran número de variables de entrada. Esto es debido a que por norma general a la hora de realizar una clasificación debemos tener un número considerable de variables para poder diferenciar unos datos de otros y cada variable representa una dimensión del espacio. Si estos datos se quieren reproducir en su entorno, nos encontraríamos con un espacio N-Dimensional, lo cual es imposible de representar. Por ello se hace necesario eliminar de alguna forma la complejidad de la representación de estos espacios dimensionales y obtener una forma en la que se quiera visualizar directamente dicho conjunto. Para conseguir esto es necesario recurrir a técnicas de reducción de dimensionalidad. Con estas técnicas se podrán trasladar los datos de un espacio N-Dimensional a un entorno en 2 dimensiones, lo que nos permitirá la visualización de los conjuntos de datos de forma física.

Hay una gran variedad de técnicas con diferentes propiedades, por ejemplo mantener las distancias entre vecinos. Esto conlleva a que cada una de las técnicas de reducción de dimensionalidad tenga unos puntos fuertes, es decir, unas ventajas para un cierto número de aplicaciones en las que podrá ser usado y por otro lado unos inconvenientes que puede convertir a la misma técnica en un método totalmente incompatible para otras. Se está en una situación en la que podemos comprobar que existen distintas técnicas de reducción de dimensionalidad que funcionan mejor o peor según el conjunto de datos del estudio y según la funcionalidad requerida, pero no existe una medida comparativa entre distintas técnicas de reducción de dimensionalidad. Por ello en este proyecto se crea una función de coste para poder realizar un estudio sobre la eficacia de cada una de las técnicas de reducción de dimensionalidad sobre distintos conjuntos de datos.

## 1.2 Objetivos

Los objetivos de este proyecto se pueden enmarcar en uno global y varios sub-objetivos.

- Objetivo global: la visualización de datos de  $N$  dimensiones en un espacio bidimensional. Conseguir unos resultados que permitan la comparativa de métodos de reducción de dimensionalidad y encontrar aquel que se adapte mejor a distintos conjuntos de datos y que mantenga en mayor medida las características del modelo inicial sobre su modelo reducido.
- Sub-objetivo 1: Para conseguir el objetivo global se hará uso de las salidas blandas de los clasificadores añadiendo la información a los datos. Dicha información adicional la obtendremos con SVM, la cual nos proporcionará la probabilidad de pertenencia de cada punto a cada una de las distintas clases. De esta forma los espacios de  $N$  dimensiones van a convertirse en espacios  $N+P$  dimensionales. Se realizará un estudio sobre cuán importante puede ser esta información adicional y en qué grado puede resultar más eficaz dando en distintas ejecuciones un peso a esas nuevas variables de entrada con un factor de ponderación denominado  $R$ .
- Sub-objetivo 2: Para poder realizar la comparativa de métodos de reducción de dimensionalidad se necesita de una función de coste que provea de resultados numéricos la eficiencia de cada una de las técnicas. Así se podrá comparar y analizar de forma objetiva los resultados.
- Sub-objetivo 3: Se quiere comparar cada una de las técnicas de reducción de dimensionalidad tanto de forma visual como de forma numérica con la función de coste. Con ello se conseguirá realizar una clasificación de eficiencia de las distintas técnicas de reducción de dimensionalidad sobre distintos conjuntos de datos.

### 1.3 Contenido de la memoria

El contenido de este proyecto se puede dividir de la siguiente forma:

- Índice de tablas y figuras

Índice con la información de la nomenclatura de las distintas figuras y tablas que aparecen durante el proyecto con la referencia de la página donde están contenidas.

- Estado del arte

En esta parte del proyecto se hace un estudio de las referencias bibliográficas hasta el momento. Es decir, se presentan los estudios realizados en el campo de investigación del proyecto y se presentan las herramientas de trabajo.

- Desarrollo del proyecto

Esta es la parte central del proyecto. En ella se describe toda la realización del mismo. Desde su estado inicial, como conseguir las herramientas de trabajo y los conjuntos de datos, hasta su estado final que será la representación gráfica de los resultados. También se describirán todos los estados intermedios y se indicarán todas las dificultades encontradas, las decisiones que se tuvieron que tomar para dirigir el proyecto en un camino u otro y la presentación de tablas y figuras que permiten una mejor comprensión de los resultados.

- Conclusiones y trabajos futuros

La parte final del proyecto hace referencia a una explicación sobre los resultados obtenidos y una descripción sobre las distintas posibilidades de seguir ampliando el trabajo realizado.

- Anexos

En estos se añadirán todas las visualizaciones de los métodos de reducción de dimensionalidad, así como las tablas que muestran los resultados obtenidos en cada técnica y en cada ejecución.

## Capítulo 2: Estado del arte

### 2.1 Introducción

Este proyecto se basa en dos software completamente diferentes. Los cuales, de forma conjunta, permitirán la buena realización del proyecto. Por ello, en este apartado separaré el estado del arte de cada uno de ellos. Por un lado tendremos la librería de SVM, la cuál permite el cálculo de distancias, la creación de modelos y la predicción de datos. Por otro nos encontraremos con las técnicas de reducción de dimensionalidad, las cuales permiten transformar el espacio  $N$  dimensional a uno bidimensional.

### 2.2 LIBSVM

LIBSVM es una biblioteca para máquinas de soporte vectorial, Support Vector Machines [1] [22]. La librería ha sido desarrollada desde el año 2000 con la finalidad de ayudar a los usuarios para aplicar fácilmente SVM a sus aplicaciones. LIBSVM ha ganado popularidad principalmente en el aprendizaje automático.

Las Máquinas de Vectores Soporte, SVM, son una técnica útil para la clasificación de datos, además de ser un método popular de aprendizaje de máquina para clasificación, regresión y otras tareas de aprendizaje. Desde el año 2000, se ha desarrollado el paquete LIBSVM, una biblioteca para máquinas de vectores soporte. La dirección web del paquete es <http://www.csie.ntu.edu.tw/~cjlin/libsvm>. LIBSVM es actualmente uno de los software SVM más ampliamente utilizado. De 2000 a 2010, había más de 250.000 descargas del paquete. En ese período, los creadores respondieron a más de 10.000 correos electrónicos de los usuarios. Aunque SVM se considera más fácil de usar que las redes neuronales, los usuarios que no están familiarizados con la técnica obtienen resultados insatisfactorios en un primer momento. Debido a ello crearon un documento para que los nuevos usuarios obtuviesen unos resultados aceptables, esta guía es “A Practical Guide to Support Vector Classification” [2].

El cuadro 2.1 enumera obras representativas de algunos dominios en los que se ha logrado utilizar LIBSVM [3].

Dominio	Trabajos representativos
Visión por computador	LIBPMK (Grauman and Darrel, 2005)
Procesamiento natural del lenguaje	Maltparser (Nivre et al., 2007)
Neuroimagen	PyMVPA (Hanke et al., 2009)
Bioinformática	BDVal (Dorff et al., 2010)

Tabla 2.1. Trabajos que han tenido éxito usando LIBSVM

El paquete [LIBSVM](#) se estructura de la siguiente manera.

1. Directorio Principal: Núcleo C / C + + programas y los datos de la muestra. En particular, el archivo svm.cpp implementa algoritmos de entrenamiento y de pruebas.
2. La herramienta sub-directorio: este sub-directorio incluye herramientas para la comprobación de datos y los parámetros para la selección de SVM.
3. Otros sub-directorios contienen archivos binarios pre-construidos y las interfaces en otros idiomas / software.

### 2.2.1 Formulaciones de SVM

En este apartado procederé a describir las formulaciones de SVM apoyadas en LIBSVM. Tanto las relativas a la clasificación, necesarias en la realización de este proyecto, como las relativas a la regresión, de las que se hará una breve mención. De esta forma nos encontramos con las siguientes posibilidades.

- Clasificación
  - 1) Apoyo-C en la clasificación de vectores (c-SVC)
  - 2) Apoyo -V en la clasificación de vectores (v-SVC)
  - 3) Distribuciones de estimación (one-class SVM)
- Regresión
  - 1) Apoyo- $\epsilon$  de vectores de regresión ( $\epsilon$ -SVR)
  - 2) Apoyo-v de vectores de regresión (v-SVR).

LIBSVM apoya diferentes métodos para la clasificación, regresión y estimaciones de distribución. Ahora se realizará un estudio conjunto de las distintas posibilidades para que resulte mucho más sencillo ver las diferencias entre cada una de las opciones que presenta SVM.

## Clasificación

Para la clasificación en LIBSVM encontramos tres tipos distintos, c-SVC, v-SVC y estimación de distribución (one clase SVM). En todos los métodos debemos introducir un vector de entrenamiento  $x_i \in \mathbb{R}^n$ ,  $i=1, \dots, l$ . En c-SVC existirán dos clases, y un indicador de vector  $y \in \mathbb{R}^l$ , tal que,  $y_i \in \{-1, +1\}$ . V-SVC introduce un nuevo parámetro  $\nu \in (0, 1]$ . Está comprobado que  $\nu$  tiene un límite superior en la fracción de errores de entrenamiento y un límite inferior de la fracción de vectores de soporte. De la misma forma en el tercer caso Schölkopf [4] propuso una clase SVM para el apoyo en las estimaciones de distribución en alta dimensión, se le da un vector  $x_i$  como en los casos anteriores con ninguna información de clase. De esta forma los problemas primigenios de optimización se resuelven de la siguiente forma

C-SVC	$\min_{w, b, \xi} \quad \frac{1}{2} w^T w + C \sum_{i=1}^l \xi_i$ <p>Sujeto a</p> $y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i,$ $\xi_i \geq 0, i = 1, \dots, l,$
V-SVC	$\min_{w, b, \xi, \rho} \quad \frac{1}{2} w^T w - \nu \rho + \frac{1}{l} \sum_{i=1}^l \xi_i$ <p>Sujeto a</p> $y_i(w^T \phi(x_i) + b) \geq \rho - \xi_i,$ $\xi_i \geq 0, i = 1, \dots, l, \quad \rho \geq 0.$
Estimacion de Distribución	$\min_{w, \xi, \rho} \quad \frac{1}{2} w^T w - \rho + \frac{1}{\nu l} \sum_{i=1}^l \xi_i$ <p>Sujeto a</p> $w^T \phi(x_i) \geq \rho - \xi_i,$ $\xi_i \geq 0, i = 1, \dots, l.$

Figura 2.1. Problemas primigenios en la clasificación de vectores

C-SVC  $\phi(x_i)$  mapea  $x_i$  dentro de un espacio de alta dimensión siendo  $C > 0$  el parámetro de regularización. Debido a la posible alta dimensionalidad del vector de la variable  $w$  para cada caso se resuelve el problema dual de la forma mostrada.



C-SVC	$\min_{\alpha} \quad \frac{1}{2} \alpha^T Q \alpha - e^T \alpha$ <p>Sujeto a</p> $y^T \alpha = 0,$ $0 \leq \alpha_i \leq C, \quad i = 1, \dots, l,$
V-SVC	$\min_{\alpha} \quad \frac{1}{2} \alpha^T Q \alpha$ <p>Sujeto a</p> $0 \leq \alpha_i \leq 1/l, \quad i = 1, \dots, l,$ $e^T \alpha \geq \nu, \quad y^T \alpha = 0,$
Estimaciones de Distribución	$\min_{\alpha} \quad \frac{1}{2} \alpha^T Q \alpha$ <p>Sujeto a</p> $0 \leq \alpha_i \leq 1/(\nu l), i = 1, \dots, l,$ $e^T \alpha = 1,$

Figura 2.2. Formulas del problema dual en la clasificación de vectores

C-SVC  $e=[1,\dots,1]^T$  es un vector de todo unos,  $Q$  es una matriz semidefinida  $l \times l$ ,  $Q_{ij} \equiv y_i y_j K(x_i x_j)$ , y  $K(x_i x_j) \equiv \phi(x_i)^T \phi(x_j)$  es la función del núcleo Después que se haya resuelto el problema utilizamos la  $w$  optima que satisfaga.

$$w = \sum_{i=1}^l y_i \alpha_i \phi(x_i)$$

 Figura 2.3. Condición para la obtención de una  $w$  óptima

En V-SVC, donde  $Q_{ij} \equiv y_i y_j K(x_i x_j)$  se demostró que el rango de uso de  $\nu$  es menor que  $(0,1]$  ya que el problema es realizable si y solo si

$$\nu \leq \frac{2 \min(\#y_i = +1, \#y_i = -1)}{l} \leq 1$$

Figura 2.4. Condición para que el problema dual sea realizable en V-SVC

Y finalmente en estimaciones de distribución  $Q_{ij} = y_i y_j K(x_i x_j) = \phi(x_i)^T \phi(x_j)$ . Después se procede a la función de decisión la cual para cada caso es

C- SVC	$\text{sgn} \left( \mathbf{w}^T \phi(\mathbf{x}) + b \right) = \text{sgn} \left( \sum_{i=1}^l y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b \right)$
V-SVC	$\text{sgn} \left( \sum_{i=1}^l y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b \right)$
Estimación de Distribución	$\text{sgn} \left( \sum_{i=1}^l \alpha_i K(\mathbf{x}_i, \mathbf{x}) - \rho \right)$

Figura 2.5. Formulas de las funciones de decisión para clasificación en SVM

En C-SVC guardamos  $y_i \alpha_i \forall_i$ ,  $b$ , los nombres de las etiquetas, la gestión de vectores y los parámetros del núcleo en el modelo de predicción. En v-SVC se muestra que  $\mathbf{e}^T \alpha \geq \nu$  puede ser sustituido por  $\mathbf{e}^T \alpha = \nu$ . Para el caso de v-SVC y de estimación de distribuciones se soluciona una versión escalada del problema dual porque numéricamente  $\alpha_i$  puede ser muy pequeño debido a la restricción de  $\alpha_i \leq 1/l$ .

V-SVC	$\begin{aligned} \min_{\bar{\alpha}} \quad & \frac{1}{2} \bar{\alpha}^T Q \bar{\alpha} \\ \text{Sujeto a} \quad & 0 \leq \bar{\alpha}_i \leq 1, \quad i = 1, \dots, l, \\ & \mathbf{e}^T \bar{\alpha} = \nu l, \quad \mathbf{y}^T \bar{\alpha} = 0. \end{aligned}$
Estimación de Distribución	$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \alpha^T Q \alpha \\ \text{Sujeto a} \quad & 0 \leq \alpha_i \leq 1, \quad i = 1, \dots, l, \\ & \mathbf{e}^T \alpha = \nu l. \end{aligned}$

Figura 2.6. Versión escalada del problema dual

Si  $\alpha$  es óptima para el problema dual y  $\rho$  es óptima para el problema primitivo, Se mostró que la  $\alpha / \rho$  es una solución óptima de C-SVM con  $C = 1 / (\rho l)$ . Así, en [LIBSVM](#), dan como resultados  $(\alpha / \rho, b / \rho)$  en el modelo.

## Regresión

De la misma forma en regresión no encontramos con dos métodos posibles por un lado tenemos  $\epsilon$ - Apoyo en la regresión de vectores ( $\epsilon$ -SVR) y v- Apoyo en la regresión de vectores (v-SVR) de los cuales solo dejo esta mención ya que en este proyecto no se hace uso de métodos de regresión de vectores.

### 2.3 Técnicas de reducción de dimensionalidad

La visualización de datos de alta dimensión es un problema importante en muchos dominios diferentes [6]. La mayoría de las técnicas no dan herramientas para representar más de dos dimensiones de datos y dejan la interpretación de datos a un observador humano. En contraste con la discusión sobre técnicas de visualización, los métodos de reducción de dimensionalidad convierten un gran conjunto de datos de alta dimensión en datos que pueden ser mostrados en un diagrama de dispersión de dos o tres dimensiones. El reto de la reducción de dimensionalidad es conservar la estructura de los datos mostrados en bajas dimensiones respecto a su estructura en alta dimensión tanto como sea posible. A continuación se presenta el gráfico que muestra las distintas técnicas para la reducción de dimensionalidad [5].

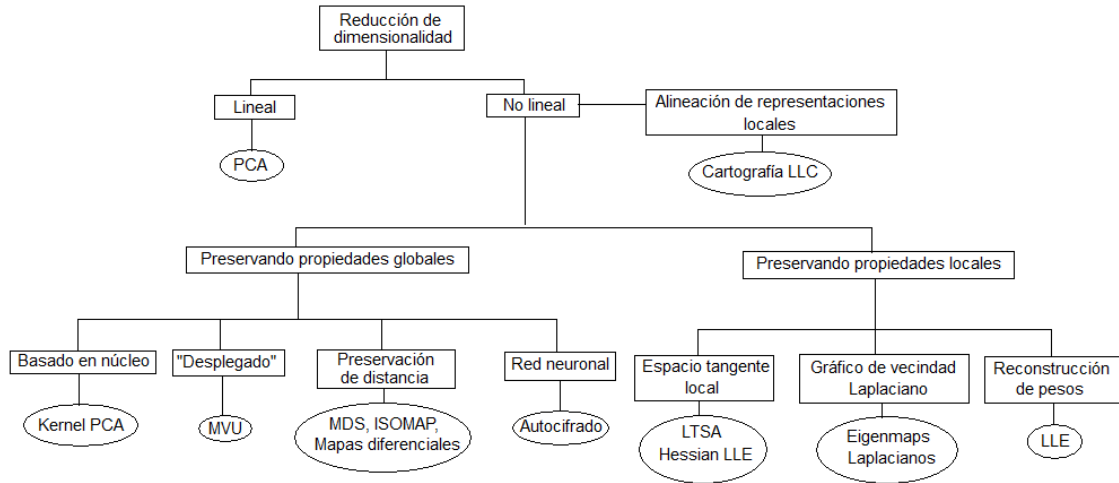


Figura 2.7. Gráfico de las técnicas de reducción de dimensionalidad

Cada una de estas técnicas presenta un conjunto de características propias para solucionar el problema de convertir los datos de alta dimensión en datos de baja dimensión. Los cuales se explican a continuación en la siguiente tabla. Donde  $n$  es el número de puntos,  $D$  la dimensionalidad original,  $d$  la dimensionalidad objetivo,  $k$  el número de vecinos cercanos,  $w$  es el número de peso en la red neuronal,  $m$  indica el número de modelos en un factor analizado de mezcla,  $p$  denota la relación de los elementos distintos a cero en una matriz de dispersión e  $i$  el número de iteraciones.

A partir de estas observaciones es claro que las técnicas no lineales imponen una demanda de recursos computacionales considerable en comparación con técnicas lineales. Los intentos de reducir la computacion y / o complejidad de memoria de técnicas no lineales se han propuesto con métodos como Isomap, MVU y KernelPCA.

El aumento del coste computacional de técnicas no lineales debe ser compensado por una mejora en el rendimiento.

Técnica	Convergencia	Parámetros	Computación	Memoria
PCA	Si	Ninguno	$O(D^3)$	$O(D^2)$
MDS	Si	Ninguno	$O(n^3)$	$O(n^2)$
Isomap	Si	K	$O(n^3)$	$O(n^2)$
MVU	Si	K	$O((nk)^3)$	$O((nk)^3)$
Kernel PCA	Si	$\kappa(-,-)$	$O(n^3)$	$O(n^2)$
Mapas de difusión	Si	$\sigma, t$	$O(n^3)$	$O(n^2)$
Autocodificadores	No	Tamaño de la red	$O(inw)$	$O(pn^2)$
LLE	Si	k	$O(pn^2)$	$O(pn^2)$
Eigenmaps Laplacianos	Si	K, $\sigma$	$O(pn^2)$	$O(pn^2)$
Hessian LLE	Si	k	$O(pn^2)$	$O(pn^2)$
LTSA	Si	k	$O(pn^2)$	$O(w)$
LLC	No	M, k	$O(imd^3)$	$O(nmd)$
Cartografía diversa	no	M	$O(imd^3)$	$O(nmd)$

Tabla 2.2. Propiedades de las técnicas de reducción de dimensionalidad

### 2.3.1 Técnicas de reducción de dimensionalidad convexas

Las técnicas para la optimización de reducción de dimensionalidad son funciones que no contienen ningún óptimo local. La solución del espacio es convexa. En estas técnicas la función objetivo tiene la forma del cociente de Rayleigh.

$$\phi(\mathbf{Y}) = \frac{\mathbf{Y}^T \mathbf{A} \mathbf{Y}}{\mathbf{Y}^T \mathbf{B} \mathbf{Y}}$$

Figura 2.8. Función objetivo de técnicas convexas

Las técnicas de espectro completo para la reducción de dimensionalidad desarrollan una descomposición de la función de la matriz completa que captura la covarianza entre dimensiones de las parejas de puntos.

## **PCA (Principal Components Analysis)/ Escalado clásico**

Es la técnica más conocida para la reducción de dimensionalidad lineal [7]. Cuyo proceso es que desarrolla la reducción insertando los datos en un sub-espacio lineal de dimensionalidad reducida que describe la varianza de los datos. Esto se hace encontrando una base lineal que reduzca la dimensionalidad de los datos donde la cantidad de la varianza sea máxima. En términos matemáticos, PCA, intenta encontrar un mapeado lineal  $M$  que maximice la función de coste. La baja representación de los datos y de los puntos  $x$  es procesada mapeándolos dentro de la base lineal.

PCA es idéntica a la técnica tradicional para el escalado multidimensional llamado escalado clásico. El escalado clásico encuentra el mapeado lineal  $M$  que minimiza la función de coste.

La función de coste es dada por la descomposición propia de la matriz de Gram  $K=XX^T$  de los datos de alta dimensión. Las entradas de la matriz de Gram se pueden obtener con la distancia euclídea al cuadrado de las parejas de puntos.

PCA y escalado clásico han tenido éxito en un gran número de campos como son reconocimiento de caras, reconocimiento de monedas y análisis de series sísmicas. Aunque tienen dos inconvenientes. Primero, en PCA el tamaño de la matriz de covarianza es proporcional a la dimensión de los puntos de datos. Como resultado, el cómputo de los vectores propios puede ser irrealizable para datos de muy alta dimensión. Segundo, la función de coste revela que PCA y escalado clásico se centran en retener las distancias de parejas grandes, en lugar de centrarse en retener las distancias de parejas pequeñas, las cuales son más importantes.

## **Isomap**

Isomap es una técnica que resuelve el problema de tomar en cuenta la distribución de los puntos vecinos [8]. Intenta preservar la distancia geodésica de las parejas de puntos. La distancia geodésica es la distancia entre dos puntos medidas en el colector. En Isomap la distancia entre puntos de datos es calculada mediante la construcción de un grafo de vecindad  $G$ , en donde todos los puntos son conectados con los  $k$  vecinos más cercanos  $x$  del conjunto de datos  $X$ . Las distancias geodésicas entre todos los puntos en  $X$  son calculadas, formando de esta manera una matriz de distancias geodésicas. La debilidad más importante de este algoritmo es su inestabilidad topológica [13]. Isomap puede construir conexiones erróneas en el grafico de vecindad  $G$ . Bastante enfoques han propuesto para superar el problema de cortocircuitado el eliminar los puntos con largos caminos de la ruta más corta del algoritmo. Su segunda debilidad radica en que Isomap sufre agujeros en el colector. Y

finalmente puede fallar para colectores no convexos. Los grafos basados en técnicas de vecindad, Isomap y LEE, no son capaces de la visualización de datos que constan de dos o más grupos ampliamente separados, ya que no da lugar a un dibujo de vecindad conectado. Se puede producir un mapa separado para cada componente conectado pero pierde la información sobre las similitudes relativas a los componentes independientes.

### **Mapas de difusión (Diffusion Maps)**

Los mapas de difusión se basan en la definición de Markov de camino aleatorio en el grafo de los datos. Realizando el camino aleatorio para el número de iteraciones, se mide la proximidad de los puntos de datos obtenidos. En la representación de baja dimensión de los datos las distancias de difusión de las parejas se retienen tan bien como es posible. La idea clave detrás de la difusión de distancias es que se basa en integrar sobre todos los caminos a través del grafo. Este método tiene aplicaciones exitosas como forma coincidente [11] y análisis de expresión genética [12].

### **Kernel PCA**

Es la reformulación del PCA lineal tradicional en un espacio de alta dimensión que es construido usando una función del núcleo. Kernel PCA calcula los vectores propios principales de la matriz del núcleo en lugar de los de la matriz de covarianza. La aplicación de PCA en el espacio del núcleo provee a Kernel PCA de la construcción de mapas no lineales. Una debilidad importante es que el tamaño de la matriz del núcleo es proporcional al cuadrado del número de instancias de la base de datos. Un enfoque para resolver este problema es propuesto en “Sparse Kernel Principal Component Analysis” [8].

### **MVU (Maximum Variance Unfolding)**

Es una técnica que intenta resolver el problema de aprendizaje de la matriz del núcleo. MVU aprende de la función del núcleo mediante la definición del grafo de vecindad sobre los datos y retiene las distancias de parejas resultantes en el grafo. MVU es diferente a Isomap en que explícitamente intenta desplegar el colector de datos. El resultado del problema de optimización puede ser resuelto usando una programación semidefinida. MVU comienza con una construcción del grafo de vecindad, en el que cada punto  $x$  es conectado a los  $k$  vecinos más cercanos. Después MVU intenta maximizar la suma de los cuadrados de las distancias euclídeas

entre todos los puntos de datos bajo la restricción de que las distancias dentro del grafo de vecindad  $G$  se preserven.

MVU tiene debilidades similares a Isomap. Los cortocircuitos pueden afectar al rendimiento por que añade restricciones para el problema de optimización que impiden un despliegue exitoso del colector. A pesar de ello MVU se usa satisfactoriamente en localización de sensores [14] y análisis de datos de microarrays [15].

### **2.3.2 Técnicas espectrales dispersas**

Las técnicas presentadas se centran en retener la estructura local de los datos.

#### **LLE (Locally Linear Embedding)**

Técnica similar a Isomap y que como ella construye un grafo de representación de puntos. Por el contrario que Isomap, intenta preservar únicamente las propiedades locales de los datos. LLE es menos sensible al cortocircuito que Isomap, porque solo un pequeño número de propiedades locales afectan si el cortocircuito ocurre. Además la preservación de las propiedades locales permite la incrustación exitosa de colectores no convexos. En la representación de baja dimensión, LLE intenta retener la reconstrucción de pesos de las combinaciones lineales tan bien como sea posible.

LLE describe las propiedades locales del colector alrededor del punto  $x$  escribiendo el punto como una combinación lineal de sus  $k$  vecinos más cercanos. El efecto de asumir la linealidad local implica que la reconstrucción de los pesos  $w$  de los puntos  $x$  es invariante a la traslación, rotación y re-escalado. Si la representación de datos en baja dimensión preserva la geométrica local del colector, la reconstrucción de pesos  $w$  que reconstruye el punto  $x$  mediante sus vecinos en la representación de los datos de alta dimensión también reconstruirá el punto  $y$  a los vecinos de la representación de datos de baja dimensión.

La popularidad de LLE ha llevado a proponer variantes lineales del algoritmo con aplicaciones exitosas como súper-resolución y localización de fuente de sonido. Los fallos de LLE son debidos a su dificultad cuando un colector contiene agujeros.

#### **Mapas propios Laplacianos**

Los mapas propios Laplacianos al igual que LLE preservan las propiedades locales del colector en la representación de datos en baja dimensión. Las propiedades locales se basan en la pareja de distancias entre vecinos cercanos. Los mapas propios

Laplacianos calculan una representación de los datos en baja dimensión donde las distancias de cada punto a sus  $k$  vecinos más cercanos son mínimas. La distancia en la representación de datos de baja dimensión entre puntos de datos al principio contribuye a la función de coste del vecino más cercano más que la distancia entre el punto y el segundo vecino más cercano. Usando la teoría de grafos espectral, la minimización de la función de coste es definida como un problema propio. Primero se construye un grafo de vecindad en donde cada punto es conectado a sus  $k$  vecinos más cercanos. Para todos los puntos  $x$  en el grafo hay conectado un borde, el peso de este borde es calculado usando la función Gausiana del núcleo. En la función de coste grandes pesos corresponden a pequeñas distancias entre puntos de alta dimensión. Esta técnica sufre la mayoría de las debilidades de LLE tales como la presencia de una solución trivial que está impedida de ser seleccionado por una restricción de covarianza que puede ser fácilmente engañada. A pesar de este fallo, es aplicado satisfactoriamente en el reconocimiento de caras [16] y en el análisis de datos fMRI [17].

### **Hessian LLE**

Es una variante de LLE que minimiza la “curviness” del colector de alta dimensión cuando lo introducimos en un espacio de baja dimensión. La curviness del colector es medido por medio de la Hessian local para cada punto. Hessian LLE comienza identificando los  $k$  vecinos más cercanos a cada punto usando distancia euclídea. La base para el espacio de la tangente local en el punto  $x$  puede ser encontrado aplicando PCA sobre los  $k$  vecinos más cercanos del punto  $x$ . Es decir, para cada punto  $x$  el espacio tangente local base es determinado calculando los vectores propios principales “ $d$ ” de la matriz de covarianza. Hay que tener en cuenta que  $k$  tiene que ser mayor que  $d$ . Este método sufre bastantes debilidades tanto de los mapas propios Laplacianos como de LLE. Una aplicación satisfactoria de este método se presentó con sensores de localización [18].

### **LTSA (Local Tangent Space Alignment)**

Análisis del espacio tangente local es una técnica que describe propiedades locales de los datos de alta dimensión usando el espacio tangente local de cada punto. Si la linealidad local del colector se asume, existe un mapeo lineal para los puntos de alta dimensión sobre ellos con el espacio tangente local y existe una lineal correspondiente a los puntos de baja dimensión en el mismo espacio tangente local. LTSA intenta alinear estos mapeos lineales de tal manera que construyan el espacio



tangente local del colector de la representación de baja dimensión. En otras palabras, LTSA hace búsquedas coordinadas simultáneamente de la representación de datos de baja dimensión y para la aplicación lineal de los puntos de datos de baja dimensión al espacio tangente local de los datos de alta dimensión. De forma similar a Hessian LLE, comienza calculando las bases de los espacios locales tangente a los puntos. Cuando esta hecho aplica PCA en los  $k$  vecinos de  $x$ . El resultado del mapeo de los vecinos de  $x$  es un espacio tangente local en el cual si existe un mapeo lineal  $L_i$  del espacio tangente local coordinado  $\theta_i$  a la representación en baja dimensión  $y_{ij}$ . Usando la propiedad de espacio tangente local y siendo  $J_k$  la matrix centrada, LTSA desarrolla la minimización siguiente

$$\min_{Y_i, L_i} \sum_i \|Y_i J_k - L_i \Theta_i\|^2$$

Figura 2.9. Función de minimización de LTSA

### 2.3.3 Técnicas de reducción de dimensionalidad no convexas

#### Sammon Mapping

El mapeo de Sammon adapta la función de coste de escalado ponderando la contribución de cada par de puntos a la función de coste con la inversa de los pares de distancias en el espacio de alta dimensión. De esta forma, la función de coste asigna aproximadamente igual peso a la retención de cada una de las distancias por parejas, y así retiene la estructura local de los datos mejor que el escalado clásico. La mayor debilidad de Sammon radica en que asigna mucho mayor peso en retener distancias de  $10^{-5}$  que retener una distancia de  $10^{-4}$ . Sus aplicaciones exitosas han sido descritas en datos geoespaciales [19] y genética de datos [20].

#### Autocodificadores Multicapa

Se basan en redes neuronales con un numero de capas ocultas y los pesos compartidos entre la capa superior e inferior. La red es entrenada para minimizar el error cuadrático medio entre la entrada y la salida de la red. Las representaciones de “y” en baja dimensión pueden ser obtenidas extrayendo valores del nodo en mitad de la capa oculta, cuando el punto  $x$  es usado como entrada. Para permitir al autocodificador aprender un mapeado no lineal entre la capa de alta dimensión y la representación de los datos en baja dimensión, generalmente se utilizan funciones sigmoideas de activación.

Este método tiene un gran número de conexiones. Por lo tanto los enfoques de propagación hacia atrás convergen lentamente y probablemente se atasquen en los mínimos locales. Este inconveniente se puede superar aprendiendo un procedimiento que se basa en tres etapas. Primero se reconocen las capas de la red y es entrenada una a una usando la Máquina de Boltzmann Restrictiva (RBM). Después, la reconstrucción de las capas de la red se forman con la inversa del reconocimiento de capas entrenadas. En la última etapa el procedimiento de entrenamiento supera la susceptibilidad de los mínimos locales con un enfoque estándar de propagación hacia atrás.

### **LLC (Local Linear Coordination)**

Coordinación Local Lineal calcula el número de modelos lineales locales y después desarrolla un alineamiento global de los modelos locales. Este proceso consiste en dos etapas, la primera una computación mixta de los modelos lineales locales en los datos por medio de un algoritmo, EM, y después un alineamiento de los modelos locales lineales obtenidos para obtener una representación de datos en baja dimensión usando una variante de LLE.

La mayor debilidad de LLC es que el montaje de la mezcla de los analizadores de los factores es susceptible a la presencia de máximos locales en la función de probabilidad logarítmica. Este método ha sido utilizado satisfactoriamente aplicado para hacer frente a imágenes de una sola persona con la variable actitud y expresión, y para dígitos escritos a mano [21].

### **Manifold charting**

El colector de gráficos construye de forma similar a LLC una representación de los datos en baja dimensión por alineamiento de un modelo MoFA o MoPPCA. Este método no minimiza la función de coste a diferencia de LLC. El método minimiza la función de coste convexa, que mide la cantidad de desacuerdo entre el modelo lineal en las coordenadas globales de los puntos. La reducción al mínimo de esta función de coste se puede realizar mediante la resolución de un problema propio generalizado.

### **SNE (Stochastic Neighbor Embedding)**

Técnica que comienza convirtiendo la alta dimensionalidad en distancias euclideas entre puntos con probabilidades condicionales que representan similitudes. Para puntos cercanos la probabilidad es relativamente alta, mientras que para puntos altamente separados es infinitesimal. Si el mapa de puntos es creado con un modelo

de similitudes de alta dimension entonces las probabilidades entre un mapa de puntos y otro serán iguales. Una medida natural de fidelidad entre los modelos de probabilidad es la divergencia de Kullback-Leibler, esta divergencia no es simétrica, diferentes tipos de error en las parejas de distancias en la dimensión reducida no están ponderados igualmente. SNE tiende a encontrar mapas con una mejor organización global. Desafortunadamente, esto requiere cambios sensibles a la cantidad inicial de ruido gaussiano y a la velocidad con la que decae.

## T-SNE

Es un método cuyo objetivo es la visualización de grandes conjuntos de datos en alta dimensión y dar una localización coordinada a coordinada en un plano bidimensional o tridimensional. Esta técnica es una variante de la “Stochastic Neighbor Embedding”, aunque mucho más fácil de optimizar y elimina la tendencia de agrupar los puntos en el centro del mapa. La función de coste de t-SNE se centra en mantener la estructura local de los datos en el mapa. La minimización de la función de coste se consigue haciendo la primera derivada. En regiones densas un pequeño valor de sigma es normalmente más apropiado que en regiones dispersas. Esta distribución tiene una entropía que aumenta si sigma aumenta. La fuerza ejercida por el resorte entre X e Y es proporcional a su longitud y también proporcional a su rigidez, que es la falta de coincidencia entre las similitudes de las parejas de puntos de datos y los puntos del mapa. En las primeras etapas de optimización, el ruido gaussiano es añadido al mapa de puntos después de cada interacción. El coste de la función en t-SNE difiere frente a la SNE en dos cosas.

- Usa una versión simetrizada de la función de coste de SNE con un gradiente simple que fue brevemente introducido por Cook.
- Usa la distribución t-Student en lugar de la gaussiana para calcular similitudes entre dos puntos en el espacio de dimensión pequeña. T-SNE emplea una distribución pesada en el espacio de baja dimensión para aliviar los cruces y los problemas de SNE.

Datos: Base de datos  $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$

Parametros de la función de coste: Perplejidad Perp

Parametros de optimización: Número de iteraciones T, velocidad de aprendizaje  $\eta$ , impulso  $\alpha(t)$

Resultado: Representación de datos en baja dimensión  $\mathcal{Y}^{(T)} = \{y_1, y_2, \dots, y_n\}$

Comienzo

Calcula parejas afines con perplejidad Perp (Figura 2.19)

introduce  $p_{ij} = \frac{p_{ji} + p_{ij}}{2n}$

la solución inicial simple  $\mathcal{Y}^{(0)} = \{y_1, y_2, \dots, y_n\}$  de  $\mathcal{N}(0, 10^{-4}I)$

para t=1 hasta T hacer

    Calcular afinidades en baja dimension

    Calcular el gradiente  $\frac{\delta C}{\delta \mathcal{Y}}$  (Figura 2.20)

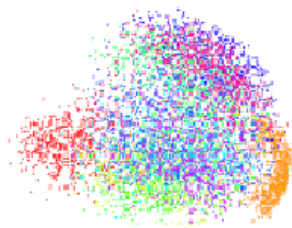
    poner  $\mathcal{Y}^{(t)} = \mathcal{Y}^{(t-1)} + \eta \frac{\delta C}{\delta \mathcal{Y}} + \alpha(t) (\mathcal{Y}^{(t-1)} - \mathcal{Y}^{(t-2)})$

fin

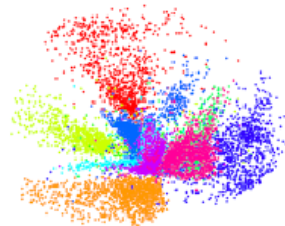
fin

Figura 2.10 Versión simple de la distribución t-SNE

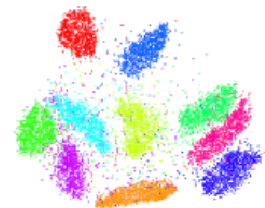
La mejor forma de comprender el mapeado de las técnicas es mediante ejemplos, por ello se presentan tres visualización diferentes.



Visualización mediante PCA



Visualización mediante Autocodificador



Visualización con t-SNE paramétrica

Figura 2.11. Visualización de 10.000 dígitos de la base de datos MNIST con técnicas paramétricas de reducción de dimensionalidad

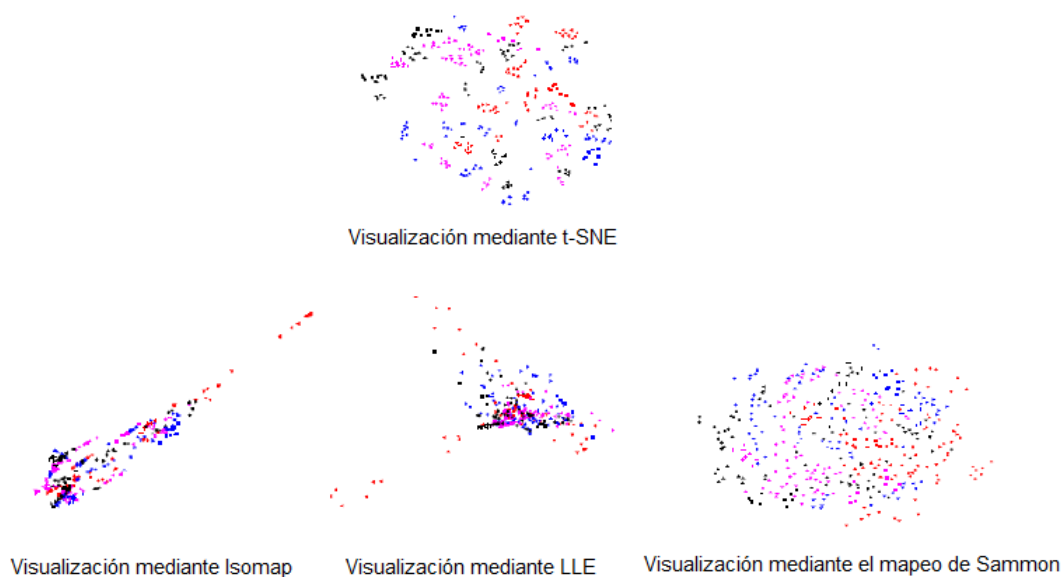


Figura 2.12. Visualización de la base de datos de las caras de Olivetti

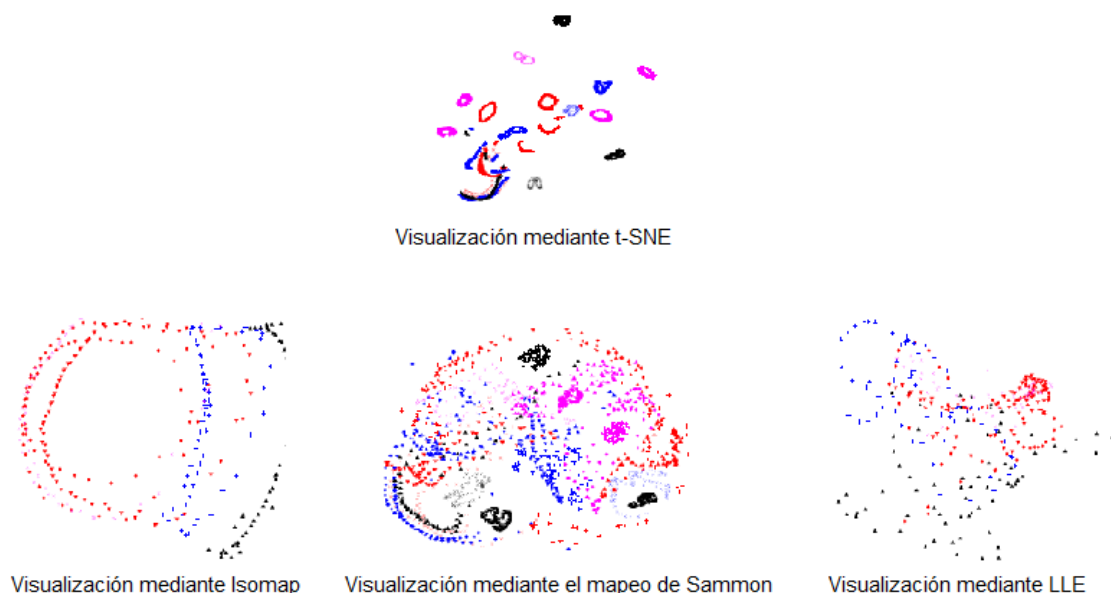


Figura 2.13. Visualización de la base de datos COIL-20

	MNIST			Characters			20 Newsgroups		
	2D	10D	30D	2D	10D	30D	2D	10D	30D
PCA	78.16%	43.03%	10.78%	86.72%	60.73%	20.50%	35.99%	27.05%	28.82%
NCA	56.84%	8.84%	7.32%	72.90%	24.68%	17.95%	<b>30.76%</b>	26.65%	26.09%
Autoencoder	66.84%	6.33%	<b>2.70%</b>	82.93%	17.91%	<b>11.11%</b>	37.60%	29.15%	27.62%
Par. t-SNE, $\alpha = 1$	<b>9.90%</b>	5.38%	5.41%	<b>43.90%</b>	26.01%	23.98%	34.30%	<b>24.40%</b>	24.88%
Par. t-SNE, $\alpha = d - 1$	<b>9.90%</b>	<b>4.58%</b>	2.76%	<b>43.90%</b>	<b>17.13%</b>	13.55%	35.10%	25.28%	<b>23.75%</b>
Par. t-SNE, learned $\alpha$	12.68%	4.85%	<b>2.70%</b>	44.78%	17.30%	14.31%	33.82%	27.21%	24.72%

Tabla 2.3. Generalización de errores de clasificadores de vecinos cercanos en representaciones de baja dimensionalidad de la base de datos MNIST, la base de datos Characters y de la base de datos 20 Newsgroups

Como se puede ver en el cuadro adjunto anterior, basan la eficacia de las técnicas de reducción de dimensionalidad en el porcentaje de datos correctamente etiquetados. Resulta que solo es una medida y para poder expresar la eficacia de la técnica necesitamos de más parámetros y no hacer exclusividad de la eficacia al porcentaje de aciertos. Es por ello que se ha realizado este proyecto y se ha creado una función de coste que permite basarse en un mayor número de parámetros.

## Capítulo 3. Desarrollo del proyecto

### 3.1 Introducción

En este apartado del proyecto se procederá a describir todo el desarrollo del proyecto desde la toma de contacto con las herramientas, los problemas surgidos durante la realización, la toma de decisiones, la creación de la función de coste y la presentación de los resultados numéricos de las distintas técnicas de reducción de dimensionalidad. Se va a presentar de forma gradual cada uno de los pasos realizados, de esta forma el lector podrá ir siguiendo de forma cronológica los pasos que se realizaron durante la realización del proyecto

### 3.2 Primeros pasos

Este proyecto se podría encuadrar en el marco de comprobación, investigación y demostración de resultados, desligándolo de la innovación. Esto es debido a que no se han creado nuevas herramientas de trabajo, sino que se ha ayudado de métodos ya desarrollados para poder obtener los objetivos marcados por el proyecto y usando estas herramientas y adaptándolas a las necesidades mostrar los resultados. Los métodos de los que me he tenido que valer han sido creados en distintos lenguajes de programación, facilitando el encuentro para un gran número de usuarios con conocimientos informáticos en distintas ramas de la programación. De entre todas las alternativas posibles mi elección fue realizarlo en el entorno de Matlab, ya que es un programa el cual he manejado durante la carrera y me resultaba más fácil a la hora de programar.

Como he dicho para la realización del proyecto necesitamos herramientas de trabajo, estas son las pertenecientes al cálculo de distancias, SVM, y las pertenecientes a la reducción de dimensionalidad.

Para obtener las relativas al cálculo de distancias, SVM, deberemos acceder a <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>, donde encontraremos gran cantidad de material. Desde aquí podremos empezar a conocer la herramienta del cálculo de distancias desde punto de vista sencillo. Ya que hay que tener en cuenta que puede ser la primera vez que trabajemos con esta herramienta necesitaremos de una explicación del método para realizar una ejecución óptima. De esta página ha sido

donde he recopilado la mayor parte de la bibliografía perteneciente a lo relativo del cálculo de distancias y donde he podido encontrar gran cantidad de ayuda con una sección de preguntas frecuentes, una guía para principiantes y el correo personal de los desarrolladores. La búsqueda del material de descarga que me interesaba, principalmente los archivos .dll, se hizo bastante difícil a través de la página web y resulto ser más sencillo a través de google. La URL de descarga de los archivos ejecutables de SVM los podemos encontrar en el siguiente enlace web <http://www.csie.ntu.edu.tw/~b91082/SVM/>. Tenemos tanto el svm-train, el cual nos sirve para el entrenamiento de los conjuntos de datos y svm-predict el cual predecirá las etiquetas de los datos y proporcionará las salidas blandas necesarias que se utilizarán como información adicional.

A la hora de comenzar a usar el programa determine leer los tutoriales y realizar primero los ejemplos propuestos para comprobar que los resultados que obtenía eran correctos y así coger la confianza para usar el software con otros conjuntos de datos. Para dar un buen uso a los ejecutables debemos de tener en cuenta las numerosas posibilidades que nos ofrece SVM ya que podemos agregar gran cantidad de parámetros para nuestro beneficio. SVM consta de las siguientes posibilidades:

- Tipo de SVM
  - C-SVC. Esta es la opción por defecto
  - $\nu$ -SVC
  - one-class SVM
  - $\epsilon$ -SVR
  - $\nu$ -SVR
- Tipo de función del núcleo
  - Lineal:  $K(x_i, x_j) = x_i^T x_j$
  - Polinómica:  $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d$ ,  $\gamma > 0$
  - Función radial base (Radial Basis Function):  $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$ ,  $\gamma > 0$ .
  - Sigmoideo:  $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$ .

En la realización del proyecto he usado las opciones por defecto, C-SVC como tipo de SVM y como función del núcleo RBF (Radial Basis Function). Esto es debido a que para principiantes C-SVC es la más recomendada y en el caso de la función del núcleo tiene la ventaja que puede manejar casos en los que la relación entre la clase



de etiquetas y atributos no es lineal. La RBF tiene menores dificultades numéricas en contraste con las polinómicas cuyos valores pueden tender hacia infinito. Este tipo de función resulta en su conjunto más completa para tratar ejemplos los cuales desconocemos su distribución. Señalar que los valores de  $\gamma$ ,  $r$  y  $d$  son los parámetros del núcleo. A la hora de ejecutar el método de cálculo de distancias podemos introducirle cierto número de parámetros con distintas características, los cuales describo a continuación. Valores pre-calculados:

- $d$ : introduce los grados en la función del núcleo.
- $g$ : introduce  $\gamma$  en la función del núcleo.
- $c$ : introduce el parámetro  $C$  de C-SVC,  $\epsilon$  - SVR, y  $\nu$ -SVR.
- $n$ : introduce el parámetro  $\nu$  de  $\nu$ -SVC, one-class SVM, y  $\nu$ -SVR.
- $p$ : introduce  $\epsilon$  en funciones de pérdida de SVR-  $\epsilon$ .
- $h$ : sirve para utilizar la heurística de contracción, 0 o 1.
- $b$ : sirve para entrenar en SVC o SVR un modelo de probabilidad
- $w_i$ : introduce el parámetro de  $C$  de la clase  $i$  al peso  $\cdot C$ , para C-SVC
- $v$ : modo de validación cruzada  $n$ -fold
- $q$ : modo silencioso

El ejemplo propuesto en el tutorial es `heart_scale`. El conjunto de datos se puede obtener en <http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html>. La sencillez de este ejemplo radica en que se encuentra un problema de clasificación binario con un gran conjunto de características, 13, lo que permite con ese conjunto de características, sin necesidad de ser pre-procesadas ya que todas son numéricas, y siendo un problema bidimensional se obtengan muy buenos resultados. Ahora bien, SVM ofrece dos posibilidades para la obtención de resultados de salida, por lo que para el objetivo del proyecto se presento el dilema de cuál de esos dos caminos sería el más conveniente. Posteriormente describiré las dos alternativas.

Teniendo nuestro primer contacto con el modelo de cálculo de distancias llega el momento de tomar contacto con los distintos modelos de reducción de dimensionalidad. Los métodos los podemos encontrar en su página oficial [http://homepage.tudelft.nl/19j49/Matlab\\_Toolbox\\_for\\_Dimensionality\\_Reduction.html](http://homepage.tudelft.nl/19j49/Matlab_Toolbox_for_Dimensionality_Reduction.html) y el material de descarga para la ejecución de los distintos métodos la obtenemos desde <http://cseweb.ucsd.edu/~lvdmaaten/dr/download.php> Con las mismas facilidades que nos encontramos en SVM nos encontramos ahora con los métodos de reducción de dimensionalidad, ya que nos encontramos con una sección de preguntas frecuentes,

así como unos documentos teóricos los cuales nos permiten una introducción hacia los métodos de reducción de dimensionalidad. Si queremos ver la ejemplificación de uno de los métodos podemos hacer uso del documento “User’s Guide for t-SNE Software” [9]. El ejemplo en esta pequeña guía propuesto es Mnist el cual se puede obtener en la página oficial de t-SNE. El principal problema con que se encuentra el usuario en la primera toma de contacto con t-SNE es el tiempo de computación con grandes conjuntos de datos, que pueden llevar en algunos momentos a la desesperación del usuario. Para hacer un buen uso de t-SNE es necesario repasar en el lenguaje de programación que se esté realizando a las funciones de representación gráfica, en mi caso gscatter y plot.

Finalmente para concluir con el apartado de presentación de las herramientas de trabajo, decir que debemos hacer uso de nuestros propios conjuntos de datos y que estos sean lo más reales posibles. Estos conjuntos se pueden obtener en el siguiente enlace <http://archive.ics.uci.edu/ml/>, gracias a la existencia de este repositorio de datos se hace posible la realización de ejemplos reales. La web cuenta con un total de 235 conjuntos de datos actualmente, los cuales vienen etiquetados con distintas características para facilitar al usuario la búsqueda de datos que necesita. Observamos que existen 7 atributos para describir las bases de datos.

- Nombre  
El nombre del archivo. Siempre nos servirá si hemos trabajado con él alguna vez y no conseguimos encontrarle en la base de datos para introducirlo en el buscador.
- Tipo de dato  
Breve descripción del tipo de datos que encontraremos en el conjunto de datos. Permitirá pasar un filtro y ahorrarnos tiempo en la búsqueda de conjuntos de datos compatibles con nuestro objetivo.
- Tarea por defecto  
Nos hace saber que uso se le puede dar a los datos, clasificación, regresión o agrupamiento entre otros.
- Tipos de atributos  
Descripción sobre los datos, estos pueden ser numéricos, descriptivos o una mezcla de ambos.

- Casos

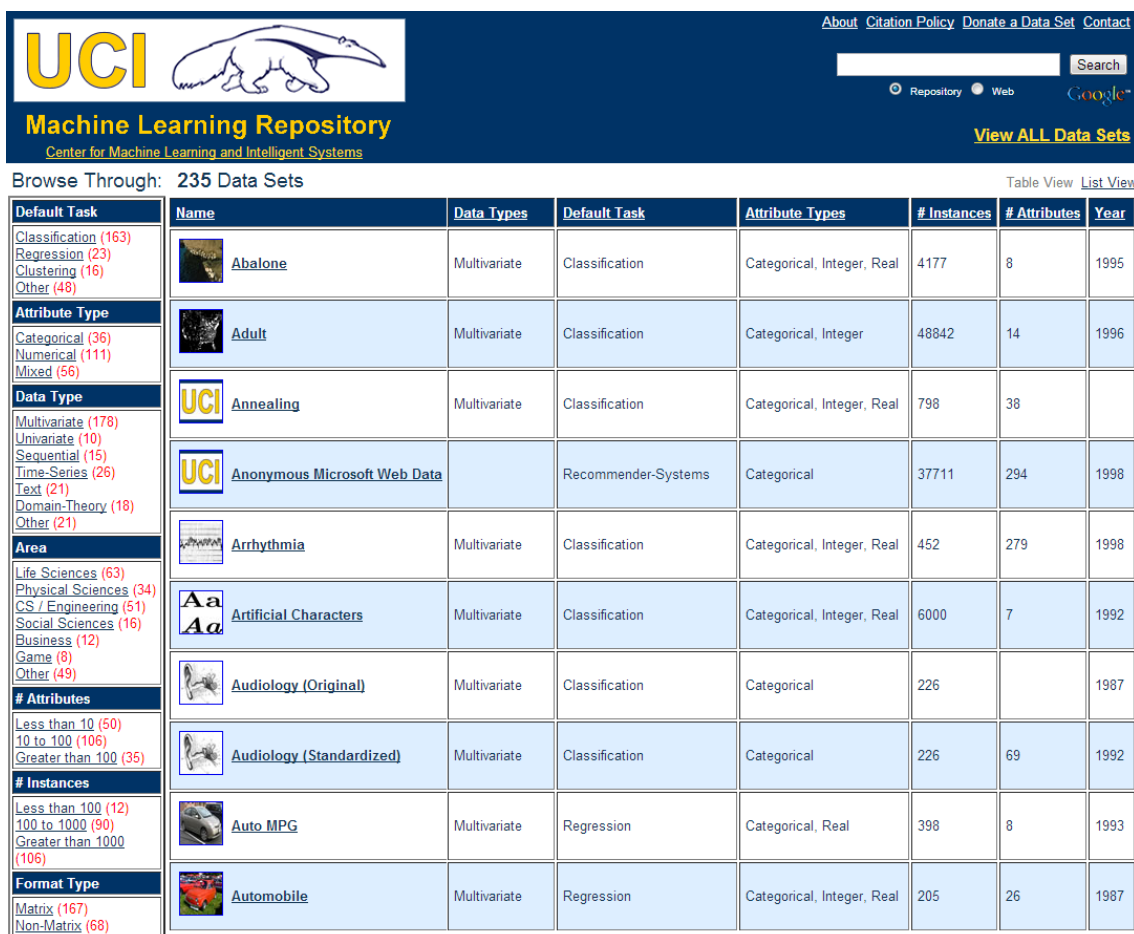
Número de elementos del que se compone el conjunto de datos

- Atributos

El número de características que tienen los datos del conjunto.

- Año

El año desde que está disponible esa base de datos.



The screenshot shows the UCI Machine Learning Repository website. The header includes the UCI logo, navigation links (About, Citation Policy, Donate a Data Set, Contact), a search bar, and a 'View ALL Data Sets' link. Below the header, there's a section 'Browse Through: 235 Data Sets' with links for 'Table View' and 'List View'. On the left, there are several filter categories: Default Task, Attribute Type, Data Type, Area, # Attributes, # Instances, and Format Type. The main table lists data sets with columns: Name, Data Types, Default Task, Attribute Types, # Instances, # Attributes, and Year.











Default Task	Name	Data Types	Default Task	Attribute Types	# Instances	# Attributes	Year
Classification (163) Regression (23) Clustering (16) Other (48)	 <a href="#">Abalone</a>	Multivariate	Classification	Categorical, Integer, Real	4177	8	1995
Attribute Type	 <a href="#">Adult</a>	Multivariate	Classification	Categorical, Integer	48842	14	1996
Categorical (36) Numerical (111) Mixed (56)	 <a href="#">Annealing</a>	Multivariate	Classification	Categorical, Integer, Real	798	38	
Data Type	 <a href="#">Anonymous Microsoft Web Data</a>		Recommender-Systems	Categorical	37711	294	1998
Multivariate (178) Univariate (10) Sequential (15) Time-Series (26) Text (21) Domain-Theory (18) Other (21)	 <a href="#">Arrhythmia</a>	Multivariate	Classification	Categorical, Integer, Real	452	279	1998
Area	 <a href="#">Artificial Characters</a>	Multivariate	Classification	Categorical, Integer, Real	6000	7	1992
Life Sciences (63) Physical Sciences (34) CS / Engineering (51) Social Sciences (16) Business (12) Game (8) Other (49)	 <a href="#">Audiology (Original)</a>	Multivariate	Classification	Categorical	226		1987
# Attributes	 <a href="#">Audiology (Standardized)</a>	Multivariate	Classification	Categorical	226	69	1992
Less than 10 (50) 10 to 100 (106) Greater than 100 (35)	 <a href="#">Auto MPG</a>	Multivariate	Regression	Categorical, Real	398	8	1993
# Instances	 <a href="#">Automobile</a>	Multivariate	Regression	Categorical, Integer, Real	205	26	1987
Less than 100 (12) 100 to 1000 (90) Greater than 1000 (106)							
Format Type							
Matrix (167) Non-Matrix (68)							

Figura 3.1. Captura de pantalla modificada del repositorio de datos

Pulsando sobre cualquiera de estas características los conjuntos de datos cambian de orden. En la parte izquierda vemos cuantos conjuntos de datos cumplen una característica en concreto y pulsando sobre la característica buscada aparecen en pantalla los elementos facilitando la búsqueda y la interacción con el usuario.

### 3.3 Aumento en la complejidad de la implementación

En este sub-apartado voy a ir describiendo los pasos que son necesarios para el buen funcionamiento del código así como las decisiones iniciales que hay que tomar para la buena realización del proyecto. Como he realizado en el apartado anterior primero comenzaré por SVM y posteriormente describiré los métodos de reducción de dimensionalidad.

En primer lugar para la realización de de SVM nos encontramos con dos formas de poder realizarlo, se verá que las formas son casi iguales, existen tan solo un par de diferencias. Estas dos posibilidades son las siguientes

1) `model=svmtrain(y,x,'-c valor -g valor');`

`[predict_label,accuracy,dec_values]=svmpredict( y, x, model);`

Por este camino el valor que vamos a recibir a la salida del estudio del conjunto de datos es `dec_values` el cual sirve para indicar los valores de decisión para una determinada variable. Estos valores eran difíciles de comprender ya que aparentemente no se regían para la decisión de la salida por ninguna regla matemática.

2) `model=svmtrain(y,x,'-c valor -g valor -b 1');`

`[predict_label,accuracy,prob_values]=svmpredict( y, x, model, -b 1);`

El parámetro `-b` como describí antes, permite convertir el conjunto en un modelo de probabilidad, por lo tanto vemos que una de las salidas de `svmpredict` se corresponde con dicha descripción, `prob_values`. Esta variable era mucho más fácil de entender que la anterior ya que a cada dato le corresponden tantas probabilidades como clases haya y ofrece una probabilidad de pertenencia con cada clase, eligiendo finalmente la probabilidad de pertenencia que fuera superior al resto. Este segundo fue el modelo elegido como el camino más correcto ya que no solo nos permitía de forma cognitiva entender los resultados que estaba mostrando, sino que nos ofrece la posibilidad de ver qué relación, o mejor dicho, a qué distancia se encontraba ese dato del resto de clases.

Tras la elección de uno de los dos modelos que se presentan debemos elegir bien los valores de las variables, ya que algunas vienen impuestas, como son el caso

de las características de clase y las etiquetas del conjunto de datos, y otras podemos manipularlas e introducirlas nosotros manualmente, las relativas a SVM. Las dos variables que podemos modificar y que influirán en el correcto funcionamiento de nuestro proyecto son  $c$  y  $g$ . Si realizásemos un estudio rápido, se escogerían los valores por defecto, estos son 1 para  $c$  y  $g$  adopta el valor de la operación de dividir 1 entre el número de características que tenga el conjunto de datos. Estas variables influirán en la creación de nuestro modelo y como consecuencia en la efectividad de nuestro decisor. Son los únicos valores que podemos modificar de una forma u otra y por ello se tiene que tener la certeza de que estos valores son los más correctos para el modelo. He aquí donde surge la necesidad de hacer un estudio profundo de estas variables y de sus valores, ya que según los desarrolladores de SVM, se mantendrían los valores recomendados por defecto para distintas bases de datos que contasen con el mismo número de características. Es decir, los mismos valores de  $c$  y  $g$  para conjuntos de datos que puedan ser de naturaleza y distribución diferente. Para el estudio de estas variables se hace uso de la validación cruzada, que nos permitirá realizar una búsqueda de los valores óptimos de estos parámetros para obtener los mejores resultados de salida.

Para realizar una validación cruzada correctamente hay que tener en cuenta ciertos factores. El primero es que dentro de nuestro conjunto total de datos, tenemos que separarlos en dos conjuntos, el de train y el de test. El de entrenamiento servirá para crear el modelo y asentar así la base y el conjunto de test para hacer la prueba de dicho modelo sobre posibles datos futuros. En este punto aparece un problema, para la validación cruzada necesitaremos un subconjunto del conjunto total, y para que este sea lo más parecido a la realidad se tiene que mantener una distribución de datos de clase que se parezca en gran medida al conjunto de datos total. Esto es un problema debido a que los conjuntos de datos se encuentran ordenados por etiquetas de clase, es decir, en un conjunto de 150 elementos si existen 50 elementos de tres clases distintas observamos que los elementos del 1 al 50 son de la clase 1, del 51 al 100 de la clase 2 y finalmente de la 101 y a la 150 de la clase 3. En este proyecto se ha escogido para todos los conjuntos de datos un conjunto de entrenamiento de dos terceras partes del total y una tercera parte como test. Si se escogen los dos tercios del conjunto descrito anteriormente como conjunto de entrenamiento no se contará con ningún elemento de la clase 3, así como en el conjunto de test solo nos encontraríamos con elementos de la clase 3. Para solucionar este problema se hace uso de la función `randperm`. Se crea una variable aleatoria que desordena los datos y de esta forma al coger 100 elementos del conjunto inicial nos encontraremos con

datos distribuidos de forma aleatoria. Teniendo un conjunto de entrenamiento que cuenta con datos de todas las clases posibles, pasaré a describir el funcionamiento de la validación cruzada.

En la validación cruzada hay que tener en cuenta que solo podemos introducir el conjunto de entrenamiento, ya que estos son los valores presentes y que permiten un estudio, mientras que los de test serán los valores futuros en los que tendremos que probar nuestro modelo para comprobar la bondad del mismo. El funcionamiento del método de la validación cruzada es sencillo. Se divide el conjunto de entrenamiento en subconjuntos de igual tamaño  $v$ . Secuencialmente un subconjunto se prueba utilizando el clasificador entrenado en los restantes  $v - 1$  subconjuntos. Por lo tanto, cada instancia del conjunto de entrenamiento se predice una vez, de modo que la precisión de la validación cruzada es el porcentaje de datos que se clasifican correctamente. En este proyecto el tamaño de  $v$  es 3, es decir, de cada tres datos dos los escoge como conjunto de entrenamiento y uno como test, al igual que he hecho en el conjunto global de datos. El funcionamiento de la validación cruzada es el siguiente. Se escogen unos rangos para los valores de  $c$  y  $g$ , estos son probados sobre el primer conjunto y probados en los restantes. Todas las pruebas se van almacenando con la eficacia de cada iteración. Posteriormente cambia los valores de  $c$  y  $g$  y los vuelve a probar sobre los conjuntos. Los valores que obtendremos al final serán aquellos que hayan dado unos mejores resultados de eficacia.

Hay que tener en cuenta que los valores de  $c$  y  $g$  no son los que aparecen en pantalla, los valores que finalmente se usaran son  $2^c$  y  $2^g$ .

La validación cruzada tiene otro problema, según se haya realizado la separación entre el conjunto de test y train en el conjunto total. Nos encontraremos con que la validación cruzada puede dar unos valores u otros ya que el conjunto de entrenamiento es distinto según la distribución que haya realizado del conjunto la función randperm. En la siguiente tabla se muestra este caso para el ejemplo wine, los distintos valores de  $c$  y  $g$  y la efectividad que nos encontramos en el conjunto de entrenamiento. Como se puede observar no se llega a ningún valor concreto, ya que con cada ejecución nos encontramos con valores distintos para  $c$  y  $g$ , así como la efectividad máxima se encuentra varios valores de  $c$  y  $g$ . Como finalmente tendremos en cuenta para medir la bonanza de los métodos de reducción de dimensionalidad tanto los valores del conjunto de train como los del conjunto de test, se decidió realizar un solo lanzamiento de randperm para el conjunto de datos ya que al tener en cuenta ambos conjuntos las diferencias posibles en la varianza de aciertos en la capa de entrenamiento podrían verse afectados de forma inversa en el conjunto de test.

Valor Calculado de C	Valor calculado de G	Tasa de acierto en entrenamiento
15.5	-20.75	100
3.5	-11	94.28
-1	-6	100
14.5	-21.5	97.14
21.5	-6.25	100
18.5	-21.5	100
21.5	-4.75	100
0	-6.75	100
21.5	-5	100
6	-11.75	100
13.5	-21	97.14
21.5	-3	100
-0.25	-6.75	100

Tabla 3.1. Valores obtenidos de los parámetros libres en varias ejecuciones del ejemplo wine

El resto de las variables con las que nos encontramos en SVM son:

- y: etiquetas de clase
- x: el conjunto de características de x
- c y g: parámetros que se obtienen con la validación cruzada.

Con la elección del método de realización y con los valores de c y g obtenidos en la validación cruzada se podría dar SVM por concluido. Mas tarde se describirá que uso le damos a la matriz de salida prob\_values.

A continuación se describe el funcionamiento de los métodos de reducción de dimensionalidad y los problemas iniciales con los que nos encontramos, al igual que se ha hecho con SVM. La línea de comando la cuál nos permitirá ejecutar las distintas técnicas de reducción de dimensionalidad sin tener que realizar grandes cambios es la siguiente: `mapped_data = compute_mapping(data, method, # of dimensions, parameters)`. Donde vemos cada uno de las distintas características.

- Data

Es el conjunto de datos de entrada

- Method

El la técnica con la que vamos a realizar la reducción de dimensionalidad,

- # of dimensions

El número de dimensiones a las que se quiera reducir el conjunto de datos, en este proyecto siempre tendrá un valor de 2.

- Parameters

Los parámetros que se pueden añadir a las distintas técnicas según su naturaleza.

Hay que tener en cuenta que para las técnicas supervisadas ('LDA', 'GDA', 'NCA', 'MCML', y 'LMNN'), las etiquetas de los datos deben estar especificadas en la primera columna de los datos usando etiquetas numéricas. Para eig\_impl se recomienda como parámetro 'Matlab' en caso de tener 10000 o menos datos y 'JDQR' para conjuntos más extensos. Debido a que nuestro trabajo no es de innovación, ni busca encontrar los mejores resultados para cada conjunto de datos con una técnica concreta, sino mostrar una comparativa entre ellas, se elegirán las opciones por defecto para las distintas técnicas. Podremos hacer uso de [ ] en el lugar asignado a parameters para cualquier técnica. De esta forma estaremos haciendo referencia que no queremos asignar unos valores concretos al conjunto, sino que queremos hacer uso de los valores por defecto que tienen asignados cada una de las técnicas de reducción de dimensionalidad. Tras esta explicación se muestran las distintas técnicas que aparecen en nuestro paquete descargado, así como las características que son necesarias para su implementación.

PCA:	- none
LDA:	- none
MDS:	- none
ProbPCA:	- <int> max_iterations -> default = 200
FactorAnalysis:	- none
GPLVM:	- <double> sigma -> default = 1.0
Sammon:	- none

Tabla 3.2. Técnicas de reducción de dimensionalidad 1 de 3



```

Isomap:      - <int> k -> default = 12
LandmarkIsomap: - <int> k -> default = 12
               - <double> percentage -> default = 0.2
LLE:         - <int> k -> default = 12
               - <char[]> eig_impl -> {'Matlab', 'JDQR'}
Laplacian:   - <int> k -> default = 12
               - <double> sigma -> default = 1.0
               - <char[]> eig_impl -> {'Matlab', 'JDQR'}
HessianLLE:  - <int> k -> default = 12
               - <char[]> eig_impl -> {'Matlab', 'JDQR'}
LTSA:        - <int> k -> default = 12
               - <char[]> eig_impl -> {'Matlab', 'JDQR'}
MVU:         - <int> k -> default = 12
               - <char[]> eig_impl -> {'Matlab', 'JDQR'}
CCA:         - <int> k -> default = 12
               - <char[]> eig_impl -> {'Matlab', 'JDQR'}
LandmarkMVU: - <int> k -> default = 5
FastMVU:     - <int> k -> default = 5
               - <logical> finetune -> default = true
               - <char[]> eig_impl -> {'Matlab', 'JDQR'}
DiffusionMaps: - <double> t -> default = 1.0
               - <double> sigma -> default = 1.0
KernelPCA:   - <char[]> kernel -> {'linear', 'poly', ['gauss']}
               - kernel parameters: type HELP GRAM for info
GDA:         - <char[]> kernel -> {'linear', 'poly', ['gauss']}
               - kernel parameters: type HELP GRAM for info
SNE:         - <double> perplexity -> default = 30
SymSNE:      - <double> perplexity -> default = 30
tSNE:        - <int> initial_dims -> default = 30
               - <double> perplexity -> default = 30
LPP:         - <int> k -> default = 12
               - <double> sigma -> default = 1.0
               - <char[]> eig_impl -> {'Matlab', 'JDQR'}

```

Tabla 3.3. Técnicas de reducción de dimensionalidad 2 de 3

NPE:	- <int> k -> default = 12
	- <char[]> eig_impl -> {'Matlab', 'JDQR'}
LLTSA:	- <int> k -> default = 12
	- <char[]> eig_impl -> {'Matlab', 'JDQR'}
SPE:	- <char[]> type -> {'Global', 'Local'}
	- if 'Local': <int> k -> default = 12
Autoencoder:	- <double> lambda -> default = 0
LLC:	- <int> k -> default = 12
	- <int> no_analyzers -> default = 20
	- <int> max_iterations -> default = 200
	- <char[]> eig_impl -> {'Matlab', 'JDQR'}
ManifoldChart:	- <int> no_analyzers -> default = 40
	- <int> max_iterations -> default = 200
	- <char[]> eig_impl -> {'Matlab', 'JDQR'}
CFA:	- <int> no_analyzers -> default = 2
	- <int> max_iterations -> default = 200
NCA:	- <double> lambda -> default = 0.0
MCML:	- none
LMNN:	- <int> k -> default = 3

Tabla 3.4 Técnicas de reducción de dimensionalidad 3 de 3

Principalmente las herramientas de reducción de dimensionalidad presentan dificultades de carácter cognitivo, es decir, presenta problemas que se deben a nuestras limitaciones como seres humanos, ya que se están presentando objetos N-dimensionales a un entorno bidimensional. Resulta imposible imaginar el entorno en alta dimensión así como aún más difícil llegar a entender la reducción punto a punto de ese espacio.

Como he dicho anteriormente estas dificultades son debidas a nuestra propia condición y a nuestros conocimientos adquiridos. Tenemos que tener en cuenta que las herramientas que estamos usando, SVM, para calcular distancias y los distintas técnicas para reducción de dimensionalidad, para realizar nuestra representación en 2D, permiten reducir la complejidad del problema inicial y nos lo presentan de una forma mucho más cognitiva.

A la hora de representar los conjuntos se presenta uno de los principales problemas de la reducción de dimensionalidad, el problema del amontonamiento, del inglés “crowding”. Considerando un conjunto de puntos que se sitúan en un espacio curvo de dos dimensiones, el cual es aproximadamente lineal en una escala pequeña y esta incrustado en un espacio de más dimensiones, es posible modelar las parejas de distancias entre puntos en el mapeo de dos dimensiones. Ahora supongamos que tiene dimensiones intrínsecas y se añade a un espacio de mayor dimensión. Hay bastantes razones por las cuales las distancias por parejas en dos dimensiones no son modelos de distancia entre puntos de un conjunto de 10 dimensiones. Por ejemplo, en 10 dimensiones, es posible tener 11 puntos que son mutuamente equidistantes y esto no es posible realizarlo fielmente en el modelo de dos dimensiones. El problema de relaciones es muy diferente entre la relación de la distribución de distancias en dos espacios. El volumen de la esfera centrada en  $i$  y escalada con  $r$  elevada a  $m$ , donde  $r$  es el radio y  $m$  la dimensión de la esfera. Así los puntos de datos que están distribuidos aproximadamente de forma uniforme en una región alrededor de  $i$  en el espacio multidimensional, al intentarlo en el modelo de distancias desde  $i$  en el espacio bidimensional nos encontramos con el problema de amontonamiento. El área en el mapa bidimensional que permite acomodar distancias de puntos no será suficiente comparado con el área que permite amoldar puntos cercanos. Por lo tanto si queremos modelar de forma precisa las pequeñas distancias, la mayoría de los puntos que están a una distancia moderada del punto  $i$ , tendrán que ser colocados bastante más lejos en el mapa bidimensional

. Un intento de abordar el problema de hacinamiento fue añadir una repulsión. La repulsión ligera es creada para introducir un modelo uniforme con una pequeña proporción de mezcla. Así que por muy distantes que sean dos puntos en el mapa,  $q_{ij}$ , nunca podrán caer por debajo de la relación  $(2p/n(n-1))$  porque el fondo uniforme esta sobre  $n(n-1)/2$  pares. Como resultado, para los puntos de datos que son lejanos en el espacio de alta dimensión,  $q_{ij}$ , será siempre mayor que  $p_{ij}$ , llevando repulsión ligera.

### 3.5 Descripción de los conjuntos de datos

Los conjuntos de datos utilizados en este proyecto han sido recogidos de <http://archive.ics.uci.edu/ml/>. Han sido utilizados un total de cuatro conjuntos de datos para realizar las pruebas y obtener los resultados. En el momento de escoger los conjuntos de datos se opto por encontrar aquellos que cumpliesen unas características

que ayudasen a mostrar los resultados del proyecto. Las características buscadas son las siguientes.

- Un número de clases razonables para que exista cierta dificultad en la separación, pero no un número demasiado elevado ya que dificulta la representación. A partir de 3 clases se encuentran buenas representaciones ya que podemos encontrar diferentes distribuciones de las clases, aunque con dos clases se puede realizar resulta poco instructivo. Por otra parte un número superior de 6 clases debido a las limitaciones de símbolos y colores de matlab dificulta mucho la interpretación.

RGB Value	Short Name	Long Name
[1 1 0]	y	yellow
[1 0 1]	m	magenta
[0 1 1]	c	cyan
[1 0 0]	r	red
[0 1 0]	g	green
[0 0 1]	b	blue
[1 1 1]	w	white
[0 0 0]	k	black

Tabla 3.5. Colores en matlab

- Características numéricas, o descriptivas que puedan transformarse a carácter numérico. Cada conjunto de datos se encuentra formado por un número determinado de clases y características de clase, es decir, todas las clases tienen el mismo número de características ya que guardan una relación entre ellas. Ahora bien, estas características pueden ser de carácter descriptivo, describe unas características, o prescriptivo, las ordena. Para que se puedan realizar los cálculos debemos de encontrar clases que tengan el mayor número de características de orden numérico, es decir, se pueda cuantificar. O en otro caso que sea una característica descriptiva convertible, es decir, que se pueda transformar en una variable numérica y mantengan la misma relación. Un ejemplo de esto sería la base de datos de car, en ella se encuentra que la descripción del vehículo es descriptiva (malo, regular, bueno y muy bueno), ahora bien podemos realizar una transformación prescriptiva de estos datos a (0, 0.33, 0.66 y 1) ya que podemos encontrar una relación entre los valores dados y

los anteriores. La máxima distancia entre clases nos la podemos encontrar entre malo y muy bueno y a su vez entre 0 y 1. El problema surge en que hay características descriptivas que no pueden realizarse de esta manera. Un ejemplo que no he utilizado en la realización de este proyecto es en el que se analizaban los ingresos de una persona según características personales. Entonces aquí surge el problema, ¿hay algún tipo de relación entre los estados de una persona, casada, soltera, divorciada, viuda y con pareja? ¿Se puede decir que pueden existir ciertas relaciones entre divorciada y viuda con casada? No se puede numéricamente representar estos datos con una relación, por lo que tenemos dos opciones, o suprimir esta característica reduciendo el número de características de la clase, lo que puede conllevar a una reducción de la tasa de acierto al eliminar una variable, o transformar los datos a carácter numérico y que la relación entre los mismos se pueda ver afectada y transforme la tasa de acierto a mejor o peor mas por una cuestión de azar que por la transformación correcta de los datos.

- Un número de elementos en el conjunto de datos suficiente para que sea instructivo. Nuestra base de datos además debe contar con un número de elementos que aporten un reto en la clasificación sin llegar a imposibilitarlo. Es decir, nuestro conjunto de datos necesita que exista un número lo suficientemente grande para que exista el reto de clasificación pero sin llegar a conjuntos que por su número de datos nos eleve el tiempo de computación a niveles prohibitivos.
- Convertir la distribución de características como una distribución gaussiana. Las clases contienen características propias, y estas pueden ser de cualquier tipo, por lo que la diferencia entre los valores numéricos entre cada característica puede ser un agravante a la hora de realizar el cálculo. Por ejemplo la vida de un árbol se puede calcular en varios cientos de años y el radio de su tronco en metros. Para evitar que ciertas características puedan solapar a otras dando más importancia a las de un valor numérico mucho más alto, ya que la hora de calcular los puntos en la dimensión reducida se efectúan cálculos cruzados, lo que hay que realizar será una normalización de todas las características de la clase. De esta forma conseguiremos que cualquier variable se convierta en otra nueva con media 0 y varianza 1. De este modo conseguiremos que todas las variables tengan un rango de valores muy parecidos favoreciendo a que los cálculos cruzados se mantengan constantes.

- La tasa de acierto tenga un mínimo del 90

Para que sea válido el estudio de este proyecto y se puedan visualizar los objetivos marcados, se necesito de unos conjuntos adaptados a las necesidades. Para que existiese una buena visualización se puso como condición que la tasa de acierto tenía que ser superior o igual al noventa por ciento. Esto es debido a que una tasa menor conllevaría gran número de errores los cuales quedarían mal clasificados, se mezclarían los datos validos con los errores y dificultarían la visualización del resultado final y la creación de fronteras. Los ejemplos usados en este proyecto tienen todos una tasa de acierto por encima de este límite, lo que conlleva que haya un número muy reducido de errores.

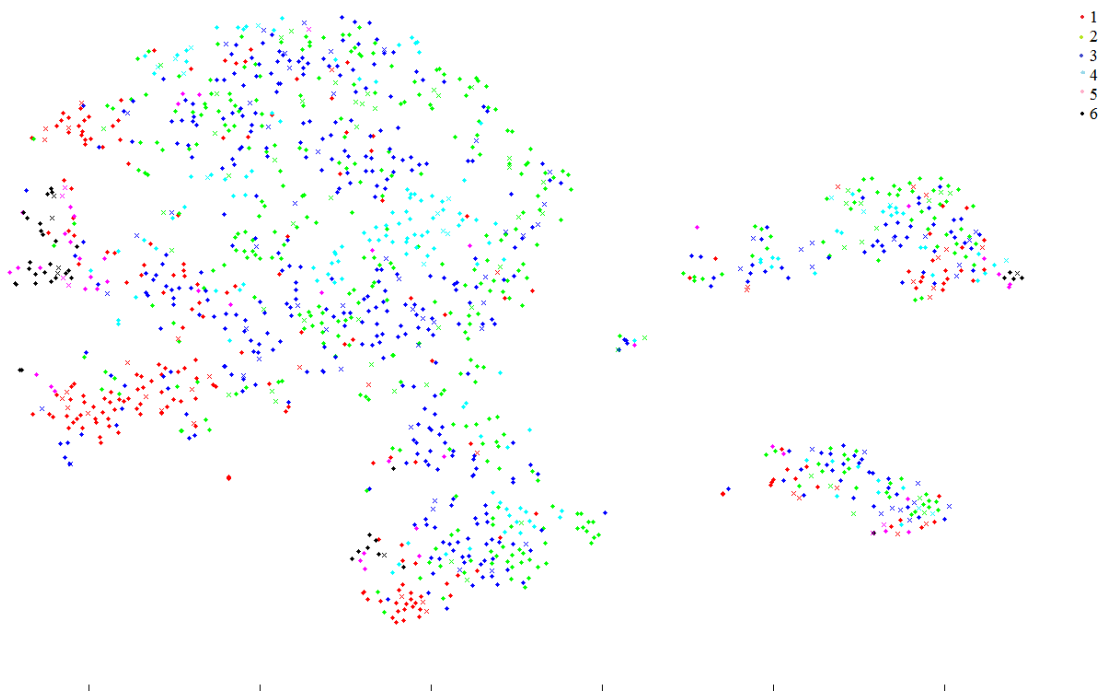


Figura 3.2. Imagen de ejemplo para un conjunto con tasa de acierto baja

En esta figura la tasa de acierto estaba en torno al sesenta por ciento, como se puede comprobar los datos están entremezclados y resulta casi imposible ubicar el centro de cada clase y la dispersión de cada una de ellas ya que los puntos se encuentran entremezclados entre sí.

Teniendo en cuenta las características que van a favorecer a la interpretación y a la buena realización del proyecto se procede a describir los conjuntos de datos seleccionados.

### 3.5.1 Conjunto Iris

Es uno de los conjuntos de datos más usado para el patrón de reconocimiento ya que es un conjunto que se puede separar linealmente. El conjunto de datos tiene 3 clases, siendo estas Iris Setosa, Iris Versicolor e Iris Virginica. Cada clase cuenta con 50 elementos.

El conjunto contiene 4 atributos, estos son:

- Sépalo largo
- Sépalo ancho
- Pétalo largo
- Pétalo ancho

Todos los datos son numéricos de carácter continuo lo que provocó que no tuviera que realizarse ningún preprocesado.

### 3.5.2 Conjunto Wine

El conjunto se basa en los resultados de un análisis químico de vinos cultivados en la misma región en Italia, pero extraídos de tres cultivos diferentes. El análisis determina las cantidades de 13 constituyentes encontrado en cada uno de los tres tipos de vinos. Tiene un total de 178 elementos distribuidos en 59 de la clase 1, 71 de la clase 2 y 48 de la clase 3.

La información de los atributos es la siguiente

- Alcohol
- Acido málico
- Ceniza
- Alcalinidad de cenizas
- Magnesio
- Fenoles totales
- Los flavonoides
- Fenoles no flavonoides
- Proantocianinas
- La intensidad de color
- Hue
- OD280/OD315 de vinos diluidos
- Prolina

Todos los atributos son continuos por lo que no fue necesario ningún tipo de preprocesado.

### 3.5.3 Conjunto balance

Este conjunto de datos fue generado para modelos de resultados experimentales psicológicos. Cada ejemplo es clasificado tomando la posición hacia la izquierda, derecha o estado de equilibrio. Los atributos son el peso a la izquierda, la distancia a la izquierda, el peso derecho y la distancia derecha. El camino correcto

para encontrar la clase sería peso izquierdo \* distancia izquierda y peso derecho \* distancia derecha. En caso de igualdad se encuentran en equilibrio.

Este conjunto tiene un total de 625 instancias, donde 49 se encuentran en estado de equilibrio (Balanced), 288 tienden a la izquierda (Left) y 288 a la derecha (Right). Este conjunto contiene cuatro atributos, la información de atributos es la siguiente:

1. Peso- Izquierda: 5 (1, 2, 3, 4, 5)
2. Distancia-Izquierda: 5 (1, 2, 3, 4, 5)
3. Peso-Derecha: 5 (1, 2, 3, 4, 5)
4. Distancia-Derecha: 5 (1, 2, 3, 4, 5)

Como se puede comprobar estos datos van a necesitar únicamente una normalización, ya que todos los atributos son numéricos. El único cambio que hay que realizar es cambiar los nombres de clase a forma numérica, convirtiendo L, B y R en 1, 2 y 3.

### 3.5.4 Conjunto Car

El conjunto de datos coches fue derivado de un modelo de decisión jerárquico desarrollado para la demostración de DEX M. Bohanec, V. Rajkovic: Expertos en sistemas de decisión de fabricación. El modelo de evaluación de coches es acorde al siguiente concepto de estructura:

- Coche: Aceptabilidad del coche
- Precio: Precio total
- Compra: Precio de compra
- Mantenimiento: Precio de mantenimiento
- Tecnología: Características tecnológicas
- Comodidad: Comodidad
- Puertas: Número de puertas
- Personas: Capacidad de personas para transportar
- Tamaño maletero: Tamaño del maletero
- Seguridad: Seguridad estimada del coche



Este conjunto tiene un total de 1728 datos donde se distribuyen los datos de la siguiente forma, 1210 inaceptables, 384 aceptables, 69 buenos y 65 muy buenos.

El conjunto tiene 6 atributos y dicha información de atributos es la siguiente

- Compra: Muy alta, alta, media, baja
- Mantenimiento: Muy alto, alto, medio, bajo
- Puertas: 2, 3, 4, 5-más
- Personas: 2, 4, más
- Maletero: pequeño, mediano, grande
- Seguridad: baja, media, alta

Como se puede comprobar todos los atributos son descriptivos o tienen un parte de ella descriptiva. Por ello se hace necesario un preprocesado. De esta forma en Mantenimiento las variables han pasado de muy alto, alto, medio bajo a (1,0.66, 0.33 y 0) y en el resto se ha realizado la misma transformación se ha pasado de variables descriptivas a variables prescriptivas para que pueden ser usadas en los cálculos.

### 3.6 Primeras visualizaciones de datos

En esta parte voy a orientar como es la metodología para conseguir los resultados mostrados. Llegados a este punto hay que suponer que tenemos un conjunto de bases de datos las cuales se han normalizado y constan de un conjunto de clases y características de clase óptimas para la visualización y para la computación. Gracias a SVM habremos conseguido la variable que nos permite conocer la probabilidad de que ese dato pertenezca a cada una de las variables. Esa nueva variable que tendrá un tamaño  $p$ , siendo  $p$  el número de clases distintas existentes en el conjunto de datos.  $P$  se añade a los datos iniciales, es decir, aumentamos en las características iniciales con  $p$  características nuevas. Procedemos a reducir su dimensionalidad con algunas de las técnicas mostradas anteriormente. Podemos comprobar que al existir un aumento de características y sobre todo a tener unos valores muy diferenciados, siendo un valor cercano a 1 la clase más probable y cercano a 0 el resto, se consigue que haya una mejor representación de los datos en dos dimensiones. Este mismo proceso se realizó varias veces pero con una única diferencia. Las condiciones de probabilidad las multiplicábamos por un factor de ponderación  $R$  comprendido entre 0 y 2, ya que se quería comprobar la influencia que podían ejercer estos nuevos valores cuando eran tenían distinto peso.

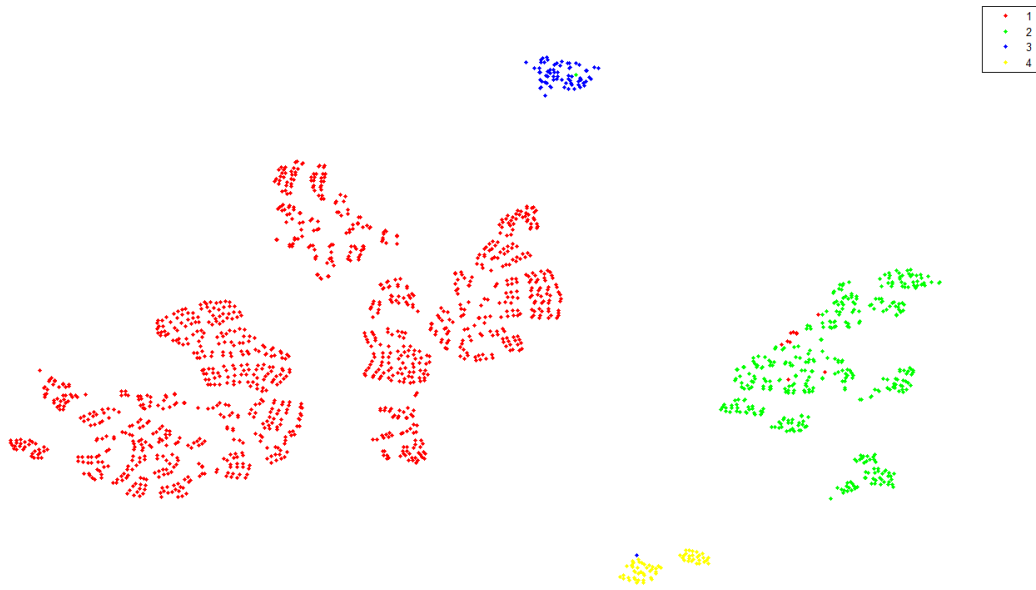


Figura 3.3 Ejemplo car para una iteración mediante t-SNE

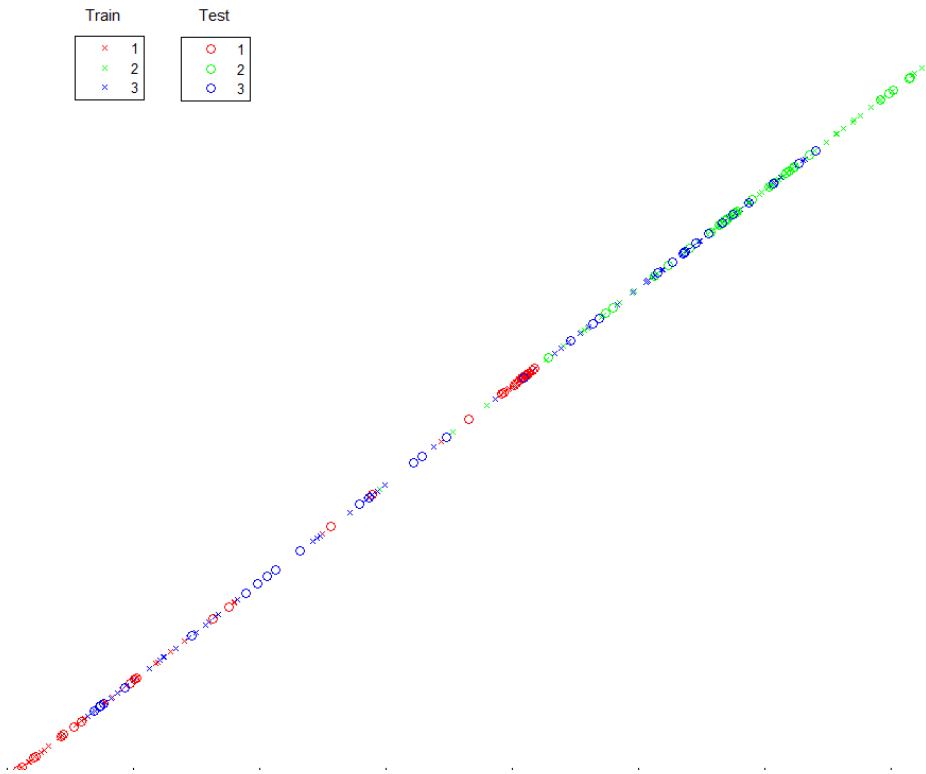


Figura 3.4 Ejemplo iris para una iteración mediante LDA

Estas visualizaciones se han realizado en su estado base, es decir se ha multiplicado la información adicional por un factor y se ha realizado la reducción de dimensionalidad. Para su representación lo único que se ha tenido en cuenta es pintar

cada una de las clases de un color diferente y denotar cada conjunto, entrenamiento y test, con un símbolo diferente.

### 3.7 Método de medición entre modelos dimensionales

Se ha conseguido reducir de dimensionalidad un conjunto de datos transformándolo mediante distintas técnicas de reducción de dimensionalidad. Se ha podido representar gráficamente y observar como los valores se separaban por clases permitiéndonos un mejor entendimiento de la distribución de los datos. Pero no se puede llegar a un acuerdo sobre la bondad de cada técnica ni se puede realizar una comparativa de técnicas solo con la representación gráfica ya que tan solo podemos dar una vaga opinión sobre como resulta una mejor separación visual de los datos. El objetivo de este apartado será proveer de un método el cual nos indique la bondad de un método para mantener una distribución bidimensional lo más parecida a su modelo N-Dimensional, así como la comparación entre distintos métodos de reducción de la dimensionalidad. Por ello me veo obligado a buscar alguna otra fórmula para que podamos indicar de forma numérica, cual de las implementaciones es mejor. Con este objetivo se procede a la realización de una función de coste que tenga en consideración distintos factores en ambos conjuntos, el de entrenamiento y test, y permita aclarar cuál de las distintas técnicas de reducción de dimensionalidad se mantiene como la mejor para cada base de datos, así como para qué factor de multiplicación  $R$  se da. Para ello se describirán cierto número de parámetros de distinta naturaleza, que de forma conjunta nos harán conocer cuál de dichas técnicas es la idónea para una base de datos.

El principal objetivo es mostrar de forma numérica la similitud entre los datos en un entorno N-Dimensional y los mismos datos en un entorno de 2-Dimensiones. Para crear un método que nos ofrezca esta posibilidad se debe tener en cuenta que deberemos de fijarnos en varias características, y que estas de forma conjunta formen el método completo. Al basarnos en más de una característica, lo que se hace es que al estar influido por varios parámetros, el error de uno de ellos solo afectara en un porcentaje del total de la función de coste. Dicha función propuesta se basa en tres parámetros por duplicado, que quiero decir con esto, estos parámetros se obtendrán tanto del conjunto de entrenamiento como del conjunto de test. Por lo que obteniendo tres parámetros de cada conjunto tenemos los seis parámetros descritos. Los parámetros utilizados son los siguientes:

- Tasa de acierto o efectividad
- Función Kendall- $\tau$
- Similitud de errores

Estos tres métodos en su primera realización se ejecutaron de forma separada, para ver los resultados y poder solucionar fácilmente los errores que cada uno de ellos pudiera ocasionar. Finalmente cuando ya todos los métodos devuelven los resultados esperados podemos realizar una sola llamada que ejecute los tres métodos de forma automática, ahorrándonos tiempo de ejecución y pudiendo observar todos los resultados de forma simultánea.

Para la realización de este método primero debemos desarrollar un estado inicial, es decir, dar unos valores base los cuales serán usados para las siguientes ejecuciones. Estos son obviamente el estado de N dimensiones. Ya que se quiere comparar la similitud de una representación bidimensional con la inicial, la representación N-Dimensional.

### 3.7.1 Estado inicial

. Lo primero que debemos hacer es normalizar todas las variables de los datos, para que sus valores tengan media 0 y varianza  $\pm 1$ . De esta forma no existe una variable que tenga un mayor peso en los cálculos. Posteriormente separaremos el conjunto de datos en un conjunto de entrenamiento y en un conjunto de test. Para la realización de este proyecto he escogido dos tercios de los datos como valores de entrenamiento y un tercio como datos de test. Los datos al estar ordenados tendremos que desordenarlos mediante la función randperm y posteriormente escoger las dos terceras partes. Así nos aseguramos que no exista una manipulación de los datos y de forma estadística se escoge un conjunto de datos con las mismas características que el conjunto completo pero en un conjunto menor. Realizo la validación cruzada de los datos en N dimensiones para obtener el mejor modelo de N dimensiones con sus valores de c y g óptimos. También se guardará su tasa de acierto tanto en train como en test.

La probabilidad en N dimensiones a priori es la máxima posible, ya que estamos dando toda la información de cada dato en un espacio ampliado. En dos dimensiones utilizaremos las probabilidades de un dato de pertenecer a cada una de

las clases posibles, por lo que como consecuencia, un error obtenido en  $N$  dimensiones debería mantenerse en dos dimensiones ya que debería tener una mayor probabilidad a una clase que no es la suya si es un error.

Para la realización del segundo parámetro descrito, Kendall- $\tau$ , se procede de la siguiente forma. Se separan los datos de entrenamiento en cada una de las clases de datos, de tal forma que tengamos agrupados todos los elementos de cada clase. Calculamos el punto medio de estos valores, ya sea en  $N$  dimensiones como posteriormente tendremos que hacer para dos dimensiones. A los datos ya agrupados los etiquetamos desde el número uno hasta el final del conjunto y así hacemos con cada conjunto de clase que hemos obtenido. Calcularemos la distancia de cada punto a la media de cada una de las demás clases y finalmente ordenaremos de menor a mayor distancia los elementos guardando como dato su etiqueta de posición anteriormente dada. De esta forma nos encontraremos con una distribución de las etiquetas anteriores regida por la distancia desde ese elemento a la media de clase de otra variable. El vector clasificado en  $N$  dimensiones será el que tendrá que ser comparado con los posteriores obtenidos en dos dimensiones.

Finalmente para nuestro último parámetro procedemos de una manera parecida a la realizada con Kendall- $\tau$ . Cada dato tiene una etiqueta que representa la posición que ocupaba en el conjunto inicial de los datos. Estudiaremos para cada dato si ha sido un error o un acierto comparando las etiquetas dadas por el decisor y las propias del elemento. Si resulta que es un error guardaremos esa etiqueta de posición para compararla con las distintas etiquetas que aparezcan en dos dimensiones. El mejor resultado posible será que en dos dimensiones tengamos como errores los mismos datos que en  $N$  dimensiones, esto significaría que al realizar nuestra transformación del espacio hemos guardado una relación con el espacio  $N$  dimensional y han sido los mismos elementos los susceptibles a error.

Con estos cálculos base podemos proceder a nuestro estudio con los diferentes métodos propuestos. Con el objetivo de obtener un único valor que nos muestre la bondad de cada método de reducción de dimensionalidad para cada ejecución realizada.

### 3.7.2 Reducción a dos dimensiones

Tras la experiencia que se obtuvo con las primeras visualizaciones, se decidió la realización de un método sistemático el cual mostrase para distintas ejecuciones un valor para posteriormente realizar la representación gráfica del mejor valor hallado. El número de ejecuciones para cada método de reducción de dimensionalidad son 21.

Cada ejecución se diferencia del resto por el valor por el que vamos a multiplicar a la información adicional, los valores de `prob_values`. Partiremos desde un estado en donde `R` multiplica a los valores de probabilidad por cero, hasta un factor de multiplicación máximo de dos. Cada salto tendrá un valor superior a los anteriores en cero coma una unidades. De tal forma encontraremos para los distintos ejemplos una tendencia de mejora o empeoramiento de los resultados según los valores, así como la posibilidad de encontrar picos de valor máximo o mínimo. Recordaré que no solo nos interesa la mayor tasa de acierto de valores positivos, también es importante que nos encontremos con que la distribución de datos sea lo más parecida a la de alta dimensión y que los errores en caso que existiesen, fuesen los mismos que los de alta dimensión. Como se ha dicho anteriormente, además de obtener un método que represente la similitud entre ambas representaciones, también se estudiará para distintas técnicas de reducción de dimensionalidad pudiendo así realizar una comparativa.

En el desarrollo de este proyecto se procederá a realizar las ejecuciones de cada uno de los métodos, se mostrarán los resultados y se indicará para cuales hemos obtenido los mejores resultados. Vista la parte de reducción de dimensionalidad, se procede a describir de una forma más completa cada uno de los parámetros seleccionados para nuestro valor final.

### **3.7.3 Tasa de acierto**

Este primer parámetro es el más sencillo de comprender. La forma de presentar la bondad del método de reducción se realiza comparando la efectividad del método en cada ejecución en dos dimensiones y posteriormente se mostraba el resultado por pantalla en forma de porcentaje. Esta forma se presentaba como muy vaga para poder realizar una explicación consistente del proceso de reducción a dos dimensiones.

Los datos, han pasado de tener  $N$  características a tener tan solo dos, por lo que al verse su información limitada, se debe suponer que sea más viable la aparición de errores. El conjunto de datos de estudio al igual que en  $N$  dimensiones, tendrá que realizar una validación cruzada para encontrar los valores que mejor se adapten al nuevo conjunto. Posteriormente podremos realizar el entrenamiento y la predicción de valores. Como se dijo anteriormente este proceso se repetirá un total de veintiún veces. Destacar que la información añadida, la correspondiente a `prob_values`, se agrega antes de la conversión de los datos a dos dimensiones. Finalmente obtendremos un vector que nos mostrará los valores de la tasa de acierto para cada

uno de los valores de R. Este método se realiza para el conjunto de entrenamiento y para el conjunto de test.

### 3.7.4 Kendall-τ

El método de Kendall-τ es una medida no paramétrica de correlación para variables no clasificadas similar a la de Sperman y al coeficiente producto-momento de Pearson. Mide la relación entre dos variables. El valor que obtenemos viene dado por distribución de los datos inicial contra los datos distribuidos en distinto orden. Estos valores se encuentran comprendidos entre dos valores, -1, siendo este datos incorrelados, +1, datos totalmente correlados y el estado intermedio 0, variables independientes. Existen dos variaciones de Kendall-τ, tau-b y tau-c. Las diferencias entre ambas es si utilizan relaciones de rango. En este proyecto no se utilizaran relaciones de rango, ya que se usará la opción por defecto. A continuación se puede ver la ecuación por la que se rige Kendall-τ.

$$\tau = \frac{C - D}{C + D} = \frac{C - D}{\frac{n \cdot (n-1)}{2}} = \frac{C - D}{\binom{n}{2}} = \frac{C - D}{\frac{n!}{2!(n-2)!}}$$

Figura 3.5 Función del cálculo de Kendall-Tau

Donde C son las parejas concordantes y D las discordantes. Una pareja de datos es concordante cuando se mantiene la relación de posición en ambas clasificaciones, es decir, si en la primera clasificación el dato a está en una posición superior a la b y esta misma relación se mantiene en la segunda clasificación. En caso que a sea superior a b en la primera clasificación y en la segunda clasificación sea mayor b que a, entonces se dice que es una pareja discordante.

- Concorde:  $x_i > x_j$  e  $y_i > y_j$  o  $x_i < x_j$  e  $y_i < y_j$
- Discorde: :  $x_i > x_j$  e  $y_i < y_j$  o  $x_i < x_j$  e  $y_i > y_j$

A continuación se describirá un ejemplo de realización de Kendall-τ.

Clasificación 1	Clasificación 2
1	1
2	3
3	6
4	2
5	7
6	4
7	5

Tabla 3.6. Ejemplo de Kendall- $\tau$

En el proyecto se obtiene como valor de salida un único valor de Kendall- $\tau$ . Este valor viene determinado por la media de distintos valores de Kendall- $\tau$  para los datos respecto a distintas clases. Nuestra primera clasificación viene determinada por los valores obtenidos en N dimensiones, y la segunda clasificación, por la clasificación en dos dimensiones. Finalmente se comparará el orden de clasificación obtenido en N dimensiones y el de dos dimensiones. Vemos como para conjuntos de datos que tengan tres clases diferentes, obtendremos seis Kendall- $\tau$  diferentes. Ya que tenemos que realizar una ordenación para cada clase respecto a las otras, de esta forma conoceremos que posición ocupaba cada punto respecto a las otras clases. Mi Kendall- $\tau$  mostrará como resultado la media de esas Kendall- $\tau$ . Ahora procedo a mostrar cómo se podría calcular manualmente este método.

1º Clasificación	2º Clasificación	C	D
1	1	6	0
2	3	4	1
3	6	1	3
4	2	3	0
5	7	0	2
6	4	1	0
7	5		

Tabla 3.7. Esquema de la realización de asignación de puntos en Kendall-Tau

Lo único que debemos hacer es sumar los C's y los D's e insertar dichos valores en la ecuación mostrada anteriormente. En este ejemplo obtenemos el siguiente resultado:



$$\tau = \frac{15 - 6}{15 + 6} = \frac{7}{21} = .42857$$

Figura 3.6 Función y resultado del ejemplo de Kendall-Tau propuesto

Al igual que con el método anterior, este procederá a realizarse un total de veintiún veces, cada una para los distintos valores de R y sobre ambos conjuntos, entrenamiento y test. La Kendall- $\tau$  como se dijo en la introducción se utilizará para conocer la correlación entre el conjunto de datos en N dimensiones y el conjunto de datos en dos dimensiones. Como nuestro objetivo es conocer que método de reducción de dimensionalidad consigue mantener el espacio más constante, es decir, la distribución de los puntos sea lo más parecida con el espacio N-Dimensional. Este parámetro se convierte en indispensable para el cálculo de la variable final.

### 3.7.5 Comparación de los errores

Este parámetro de medición consiste en comprobar si los errores que se comenten en el espacio N-dimensional son los mismos que se provocan en el espacio de dimensión reducida. Tenderemos a pensar que esto debe de ser así ya que los puntos que estén ubicados por sus características más cercanos a otras clases que a la propia en el espacio N-dimensional obtendrán una probabilidad mayor para la clase errónea. De esta forma al añadir información adicional que incrementa el error, se debería de seguir manteniendo en dos dimensiones. Los puntos donde puede existir una mayor variación entre acierto y error serán en los puntos frontera ya que estos son susceptibles de cambiar de una clase a otra.

Como se describió en el estado inicial, tenemos un conjunto de errores en N dimensiones y la etiqueta de la posición que ocupaba ese valor. En dos dimensiones deberemos realizar el mismo proceso que en N dimensiones. Buscaremos tras realizar la tasa de acierto todos aquellos valores que han sido un error y guardaremos su etiqueta de posición. Posteriormente se procede a comprobar si las etiquetas de errores entre en conjunto de N dimensiones y el de dos dimensiones son iguales. En caso de ser así diremos que existe un error coincidente. Para conseguir un valor de salida que pueda ser cuantificado se optó por realizar lo siguiente. Se crea un método el cual compare las etiquetas de error en dos dimensiones con la de N dimensiones, por cada valor que sea igual al de N dimensiones se añade un valor 1. Después se eliminan los ceros, donde las etiquetas no son iguales, y se procede a calcular el tamaño del vector que en el mejor de los casos será del mismo tamaño que en N dimensiones, es decir, contendría los mismos errores que en espacio N Dimensional.

La dificultad de este método radica en si no hay existencia de errores en el espacio N-Dimensional.

### **3.7.6 Valor de salida**

Con los tres métodos propuestos hemos obtenido ciertos valores de salida, pero se necesitan agrupar todos estos valores para crear uno solo que nos ayude a entender cuál de los métodos de reducción de dimensionalidad ha sido el mejor y para que valor de R se ha dado la distribución más parecida al espacio inicial. Como cada uno es de naturaleza distinta, ya que uno da efectividad, otro la medición de Kendall- $\tau$  que tiene valores entre -1 y 1 y los errores coincidentes que es un número entero. Se debe crear la forma que todos los parámetros aporten el mismo grado de valor. Se decidió que para entender mejor el resultado este tendría que tener unos valores de salida entre 0 y 1, siendo 1 el mejor resultado posible, es decir se asemeja nuestro modelo en dimensión reducida al original. Por ello estandarizaremos los valores de nuestros parámetros para que en su conjunto den un valor entre 0 y 1. Teniendo en cuenta que tendremos seis parámetros distintos hay que tener en cuenta que lo siguiente descrito vendrá dividido entre 6 ya que cada uno tendrá un valor máximo de 1 y tiene que aportar tan solo una sexta parte del valor total.

### **Tasa de acierto**

La tasa máxima de acierto viene dada en porcentaje, es decir en tantos por cien, por lo que lo único que se tiene que hacer es dividir la tasa de acierto de nuestro conjunto de entrenamiento o el de test por cien.

### **Kendall- $\tau$**

Sus valores oscilan entre  $\pm 1$ , por lo que para obtener el valor deseado lo que se realiza es al valor de salida de nuestra Kendall- $\tau$  sumarle 1. Esto provoca que los nuevos valores oscilen entre 0 y 2, por lo que lo único que hay que hacer ahora es dividir entre dos. De esta forma si se diera el peor caso, -1, la salida sería 0 y en el caso mejor, +1, tendríamos como salida 1. Este valor como se dijo anteriormente se divide por 6 para representar su aportación al total.

### **Errores coincidentes**

Se obtiene la longitud del vector de N dimensiones y lo sumamos a la longitud del vector de 2 dimensiones y se divide su resultado por el doble de la longitud del vector de N dimensiones. En caso que en dos dimensiones y N-Dimensiones se dé el

mismo número de errores este obtendrá el mejor valor posible para este parámetro de medición ya que cumpliría con el objetivo, tener los mismos errores en ambos espacios dimensionales dando una división de 1. Ahora bien, si resulta que en alguno de los conjuntos no existen errores hubo que tomar la decisión de cómo dar un valor de salida, ya que si hacemos el procedimiento anterior nos podemos encontrar con que el resultado sea 0/0 o un número entero entre 0. Se procedió a crear un conjunto de excepciones, en donde si en el conjunto N dimensional contaba con 0 errores y de la misma forma existían 0 errores en el espacio bidimensional la salida fuese 1, ya que tenemos el mismo número de errores en los distintos espacios dimensionales. En caso de tener 0 errores en el espacio N-Dimensional y tener alguno en el espacio bidimensional, la salida sería 0, ya que no se parece al conjunto original. Puede parecer muy radical esta decisión, ya que no sería lo mismo el caso en que nos encontrásemos con ningún error en N dimensiones y tengamos dos o diez errores en un espacio bidimensional, pero la proliferación de más o menos errores ya está presente en la efectividad, por lo que tomar otro tipo de decisión haría que esa información apareciese por duplicado.

### 3.8 Resultados obtenidos

En la siguiente sección se muestran los resultados que se han obtenido para las distintas bases de datos. Se muestran las diferentes técnicas en las que se han obtenido resultados, marcando para cada método el mejor caso e indicando para qué métodos ha resultado imposible la implementación. Debido a que es un trabajo de investigación y comprobación se ha procedido a la ejecución de las distintas técnicas de reducción de dimensionalidad, en aquellas en las que se ha dado algún error que impida la implementación no se ha realizado un estudio profundo para comprobar y solucionar dicho problema. Para los métodos en los que obtenemos resultados se coloca a la izquierda la ejecución y en el resto de columnas el método con cada uno de los resultados obtenidos.

### 3.8.1 Ejemplo iris

- Métodos para los cuales no se han obtenido resultados

FactorAnalysis	Sammon	Isomap	FastMVU
HessianLLE	LTSA	MVU	CCA
ManifoldChart	LLTSA	SPE	LLC
LandmarkMVU	GDA	LMNN	CFA
LandMarkIsomap	LLE	Laplacian	NPE

Tabla 3.8. Métodos para los que no se han obtenido resultados en iris

- Métodos para los que se han obtenido resultados

		PCA	MDS	ProbPCA	GPLVM	LPP	KernelPCA
R	0	0.6785	0.6785	0.6680	0.6785	0.6757	0.7169
	1	0.6594	0.6594	0.6618	0.6594	0.7310	0.6537
	2	0.6595	0.6595	0.6713	0.6595	0.6955	0.6639
	3	0.6401	0.6401	0.6440	0.6401	0.6904	0.6573
	4	0.6379	0.6379	0.6851	0.6379	0.7536	0.6625
	5	0.6398	0.6398	0.7220	0.6398	0.6999	0.6791
	6	0.7197	0.7197	0.6284	0.7197	0.7473	0.6747
	7	0.6607	0.6607	0.7076	0.6607	0.7115	0.7308
	8	0.6225	0.6225	0.6998	0.6225	0.7077	0.7439
	9	0.6847	0.6847	0.6990	0.6847	0.7093	0.7535
	10	0.6850	0.6850	0.6442	0.6850	0.7094	0.7535
	11	0.6841	0.6841	0.6738	0.6841	0.7090	0.7591
	12	0.6871	0.6871	0.6743	0.6871	0.7483	<b>0.8023</b>
	13	0.6857	0.6857	0.6817	0.6857	0.7656	0.7426
	14	0.6895	0.6895	0.6253	0.6895	0.7486	0.7462
	15	0.6711	0.6711	0.6931	0.6711	0.7677	0.7487
	16	<b>0.7530</b>	<b>0.7530</b>	0.6932	<b>0.7530</b>	<b>0.7698</b>	0.7682
	17	0.7169	0.7169	0.7164	0.7169	0.7665	0.7687
	18	0.7403	0.7403	0.7425	0.7403	0.6363	0.7687
	19	0.7174	0.7174	<b>0.7435</b>	0.7174	0.7101	0.7682
	20	0.6986	0.6986	0.6974	0.6986	0.7110	0.7830

Tabla 3.9. Resultados para el conjunto de datos iris 1 de 3

		SNE	SymSNE	tSNE	DiffusionMaps	Autoencoder
R	0	0.6348	0.5475	0.6702	0.6402	0.6875
	1	0.6421	0.5232	0.6902	0.6785	0.6538
	2	0.7118	0.4956	0.7856	0.6611	0.6904
	3	0.7939	0.5508	0.8097	0.6440	0.6831
	4	0.7872	0.4682	0.7330	0.6427	0.6931
	5	0.7836	0.5263	0.7506	0.6462	0.6889
	6	0.7931	0.5236	0.6924	0.6430	0.6912
	7	0.8110	0.5129	0.7694	0.6472	0.6871
	8	0.7930	0.5321	0.7724	0.6655	<b>0.7323</b>
	9	0.7544	<b>0.6540</b>	<b>0.8146</b>	0.6550	0.6902
	10	0.7899	0.5012	0.7159	0.6730	0.6941
	11	0.7865	0.5582	0.8104	0.6726	0.6949
	12	0.8104	0.5396	0.8121	0.6872	0.7070
	13	0.8106	0.5710	0.8040	0.6789	0.7265
	14	0.7824	0.5747	0.7963	0.6916	0.6915
	15	<b>0.8176</b>	0.5528	0.8094	0.6730	0.7159
	16	0.7695	0.5859	0.8129	0.7127	0.7065
	17	0.7764	0.5282	0.8133	<b>0.7171</b>	0.6804
	18	0.7981	0.5790	0.8075	0.7168	0.7023
	19	0.7722	0.4999	0.8144	0.6923	0.7154
	20	0.8116	0.5684	0.7334	0.7164	0.6980

Tabla 3.10. Resultados para el conjunto de datos iris 2 de 3

		Laplacian	MCML	LDA	NCA
R	0	0.6841	0.7060	0.7227	0.7322
	1	0.6827	<b>0.8025</b>	0.6757	0.7322
	2	0.6859	0.7479	0.5971	0.6505
	3	0.7203	0.7644	0.8142	0.6510
	4	0.6836	0.7661	0.7897	0.7030
	5	0.6868	0.7196	0.6790	0.7473
	6	0.6857	0.7667	0.7961	0.7482
	7	0.6630	0.7678	0.7265	0.7474
	8	0.6671	0.7696	0.7718	<b>0.7506</b>
	9	0.6868	0.7708	0.8139	0.7209
	10	0.6350	0.7721	0.8134	0.6492
	11	0.6688	0.7738	0.7265	0.7431
	12	0.7487	0.7745	0.6790	0.7425
	13	0.6443	0.7750	0.8139	0.6408
	14	0.7293	0.7730	0.7265	0.7331
	15	<b>0.7541</b>	0.7730	0.8139	0.7457
	16	0.7369	0.7728	<b>0.8154</b>	0.7186
	17	0.7380	0.7711	0.7734	0.7451
	18	0.7207	0.7718	0.8139	0.7458
	19	0.7442	0.7706	<b>0.8154</b>	0.7457
	20	0.6983	0.7196	0.6724	0.7479

Tabla 3.11. Resultados para el conjunto de datos iris 3 de 3

### 3.8.2 Ejemplo wine

- Metodos para los cuales no se han obtenido resultados

FactorAnalysis	Isomap	SPE
HessianLLE	LTSA	CCA
LandmarkMVU	LLTSA	MVU
LandMarkIsomap	LMNN	LLC
GDA	CFA	ManifoldChart
FastMVU	LLE	

Tabla 3.12. Métodos para los que no se han obtenido resultados en wine

- Métodos para los cuales se han obtenido resultados

		PCA	MDS	ProbPCA	GPLVM	LPP	KernelPCA
R	0	0.8072	0.8072	0.8107	0.8072	0.7378	0.8037
	1	0.8072	0.8072	0.8107	0.8072	0.7370	0.8064
	2	0.8074	0.8074	0.8107	0.8074	0.7370	0.7996
	3	0.8073	0.8073	0.8105	0.8073	0.7397	0.7998
	4	0.8072	0.8072	0.8149	0.8072	0.7393	0.8102
	5	0.8072	0.8072	0.8151	0.8072	0.7394	0.8100
	6	0.7295	0.7295	0.8152	0.7295	0.7402	0.7955
	7	0.8116	0.8116	0.8151	0.8116	0.7400	0.8072
	8	0.8104	0.8104	0.8153	0.8104	0.7466	0.8055
	9	0.8121	0.8121	0.8153	0.8121	0.7466	0.8012
	10	0.8122	0.8122	0.8169	0.8122	<b>0.8283</b>	0.8071
	11	0.8145	0.8145	0.8164	0.8145	0.8281	0.8145
	12	0.8103	0.8103	0.8166	0.8103	0.8271	0.8102
	13	0.8104	0.8104	0.8193	0.8104	0.8222	0.7340
	14	<b>0.8163</b>	<b>0.8163</b>	0.8192	<b>0.8163</b>	0.8224	0.7343
	15	<b>0.8163</b>	<b>0.8163</b>	0.8193	<b>0.8163</b>	0.8228	0.7374
	16	0.8103	0.8103	0.8183	0.8103	0.8190	0.7374
	17	0.8155	0.8155	0.8179	0.8155	0.8199	0.7375
	18	0.8152	0.8152	0.8186	0.8152	0.8153	<b>0.8180</b>
	19	0.8150	0.8150	0.8217	0.8150	0.8214	<b>0.8180</b>
	20	0.8156	0.8156	<b>0.8224</b>	0.8156	0.8180	<b>0.8180</b>

Tabla 3.13. Resultado para el conjunto de datos wine 1 de 3

		SNE	SymSNE	tSNE	DiffusionMaps	LDA	Autoencoder
R	0	0.8174	0.6390	0.7395	0.7339	0.7454	0.7062
	1	0.7341	0.5332	0.7315	0.8130	0.8136	0.7274
	2	0.8192	0.5481	0.8175	0.8101	0.8140	0.8122
	3	0.8216	0.6383	0.8251	0.8102	0.8107	0.7233
	4	0.8096	0.6973	0.7413	0.8145	<b>0.8254</b>	0.7373
	5	0.8191	0.5508	0.7347	0.8146	0.8141	0.8090
	6	0.8139	0.6381	0.8211	0.8148	0.8107	0.8008
	7	0.8212	0.5585	0.8215	0.8148	0.8107	0.7325
	8	0.7442	0.6327	0.8210	0.8144	0.8150	0.8149
	9	0.8166	0.5453	0.8180	0.8150	0.8107	0.7318
	10	0.8233	0.6183	0.8261	0.8189	0.8141	0.8260
	11	0.8206	0.6307	0.8244	0.8178	0.8107	0.7345
	12	0.8285	0.6379	0.8290	0.8190	0.8107	0.7404
	13	0.8296	0.6338	0.8256	0.8187	0.8141	0.8161
	14	<b>0.8298</b>	0.6322	0.8284	0.8180	0.8107	0.7260
	15	0.8294	0.5234	0.8287	0.8180	0.8141	0.8253
	16	0.8205	0.6663	0.8244	0.8183	0.8145	0.8159
	17	0.8212	0.5614	0.8238	0.8187	0.8107	0.8123
	18	0.8265	0.6224	0.8279	0.8192	0.8120	0.8164
	19	0.8265	<b>0.7167</b>	0.8293	0.8225	0.8097	<b>0.8276</b>
	20	0.8206	0.6195	<b>0.8336</b>	<b>0.8229</b>	0.8141	0.7120

Tabla 3.14 Resultados para el conjunto de datos wine 2 de 3



		Sammon	Laplacian	NPE	MCML	NCA
R	0	0.8071	0.8138	0.7958	0.8071	0.7414
	1	0.8135	0.8139	0.7226	0.7434	0.8206
	2	0.7959	0.8112	0.7245	0.7436	0.7357
	3	0.8076	0.8142	0.7281	0.8243	0.7476
	4	0.8159	0.8152	0.7244	0.8241	0.7449
	5	0.8166	0.8154	0.7200	0.8252	0.7453
	6	0.8115	0.8154	0.7207	0.8258	0.7412
	7	0.8174	0.8156	0.7354	0.8277	0.7452
	8	0.8194	0.8154	0.7385	0.8295	0.7444
	9	0.8140	0.8152	0.7729	0.8290	0.7440
	10	0.8138	0.8164	0.7765	0.8310	0.7490
	11	0.8161	0.8167	<b>0.8053</b>	0.8309	0.7466
	12	0.8181	0.8177	0.7045	<b>0.8325</b>	0.7443
	13	0.8173	0.8193	0.7075	0.8307	0.7431
	14	0.8203	0.8190	0.7084	0.8324	0.7492
	15	<b>0.8224</b>	0.8190	0.7385	0.8316	0.7463
	16	0.7293	0.8193	0.7072	0.8288	<b>0.8235</b>
	17	0.8204	0.8190	0.7023	0.8301	0.7474
	18	0.8212	0.8196	0.7007	0.8300	0.7452
	19	0.8148	0.8198	0.7015	0.8281	0.7492
	20	<b>0.8224</b>	<b>0.8232</b>	0.6994	0.7479	0.7525

Tabla 3.15 Resultados para el conjunto de datos wine 3 de 3

### 3.8.3 Ejemplo car

- Métodos para los que no se han obtenido resultados

FactorAnalysis	Isomap	SPE	SymSNE
HessianLLE	LTSA	CCA	NPE
LandmarkMVU	LLTSA	MVU	LPP
LandMarkIsomap	LMNN	LLC	LLE
GDA	CFA	ManifoldChart	FastMVU

Tabla 3.16. Métodos para los que no se han obtenido resultados en car

- Métodos para los que se han obtenido resultados

		PCA	MDS	ProbPCA	GPLVM	KernelPCA
R	0	0.7497	<b>0.8081</b>	0.7487	<b>0.8087</b>	0.7891
	1	0.7722	0.7722	0.7487	0.7720	0.7866
	2	0.7734	0.7734	0.7589	0.7734	0.7947
	3	0.7669	0.7669	0.7638	0.7669	0.7991
	4	0.7699	0.7699	0.7887	0.7699	0.8210
	5	0.7641	0.7641	0.7759	0.7641	0.8228
	6	0.7740	0.7740	0.7819	0.7740	0.8240
	7	0.7774	0.7774	0.7909	0.7774	0.8190
	8	0.7628	0.7628	0.7945	0.7628	0.8025
	9	0.7708	0.7708	0.7969	0.7708	0.8157
	10	0.7844	0.7844	0.7888	0.7844	0.8059
	11	0.7844	0.7844	0.7797	0.7844	0.7958
	12	0.7718	0.7718	0.7866	0.7718	0.8140
	13	0.7788	0.7788	0.7968	0.7788	0.8070
	14	0.7855	0.7855	0.7816	0.7855	0.8204
	15	<b>0.7858</b>	0.7858	<b>0.8017</b>	0.7858	0.8227
	16	0.7842	0.7842	0.7813	0.7842	0.8230
	17	0.7816	0.7816	0.7835	0.7816	0.8298
	18	0.7835	0.7835	0.7878	0.7835	0.8297
	19	0.7777	0.7777	0.7940	0.7777	0.8304
	20	0.7741	0.7741	0.7884	0.7741	<b>0.8310</b>

Tabla 3.17. Resultados para el conjunto de datos car 1 de 3

		SNE	tSNE	DiffusionMaps	LDA
R	0	0.7514	0.7877	0.7562	0.7832
	1	0.7774	0.7712	0.7904	0.8175
	2	0.7664	0.7691	0.8031	0.8175
	3	0.7902	0.7948	0.7867	0.8229
	4	0.7602	0.7894	0.7844	0.8175
	5	0.7816	0.7787	<b>0.8232</b>	0.8226
	6	0.8004	0.8021	0.8045	0.8209
	7	0.8096	0.7951	0.7985	0.8175
	8	0.8120	0.7892	0.7899	0.8175
	9	0.8047	0.8062	0.7753	0.8208
	10	0.8032	0.8154	0.7701	<b>0.8306</b>
	11	0.8103	0.8046	0.7702	0.8175
	12	0.8161	0.8135	0.7703	0.8208
	13	0.8191	0.8218	0.7851	0.8175
	14	0.8168	0.8128	0.7839	0.8175
	15	<b>0.8247</b>	0.8142	0.7922	0.8244
	16	0.8194	0.8138	0.7842	0.8175
	17	0.8221	0.8207	0.7924	0.8197
	18	0.8217	<b>0.8242</b>	0.8011	0.8208
	19	0.8231	0.7908	0.8015	0.8212
	20	0.8220	0.8179	0.8089	0.8235

Tabla 3.18. Resultados para el conjunto de datos car 2 de 3

		Laplacian	Autoencoder	MCML	NCA
R	0	0.7799	0.6094	0.7396	0.7718
	1	0.7887	0.7196	0.7858	0.7550
	2	0.7974	0.7225	0.8022	0.7793
	3	0.7997	0.7412	0.8082	0.7803
	4	0.7975	0.7383	0.8228	0.7746
	5	0.8058	0.7266	0.8223	0.7748
	6	<b>0.8129</b>	0.7556	0.8224	0.7840
	7	0.8126	0.7619	<b>0.8277</b>	<b>0.8350</b>
	8	0.7951	0.7575	0.8242	0.7898
	9	0.7966	0.7213	0.8193	0.8058
	10	0.7974	0.7327	0.8124	0.7951
	11	0.8069	0.7458	0.8118	0.7842
	12	0.7948	0.7543	0.8118	0.8328
	13	0.7879	0.7333	0.8265	0.8094
	14	0.8021	0.7473	0.8052	0.7927
	15	0.7832	0.7286	0.8243	0.8161
	16	0.7828	0.7794	0.8238	0.8112
	17	0.7897	0.7905	0.8026	0.7801
	18	0.7899	0.8004	0.8154	0.7922
	19	0.7991	<b>0.8043</b>	0.8153	0.8304
	20	0.7668	0.7919	0.8152	0.8336

Tabla 3.19. Resultados para el conjunto de datos car 3 de 3

### 3.8.4 Ejemplo balance

- Métodos para los que no se han obtenido resultados

FactorAnalysis	Isomap	SPE	GDA
HessianLLE	LTSA	CCA	CFA
LandmarkMVU	LLTSA	MVU	ManifoldChart
LandMarkIsomap	LMNN	LLC	LPP
FastMVU	LLE	LDA	

Tabla 3.20. Métodos para los que no se han obtenido resultados en balance

- Métodos para los cuales se han obtenido resultados

		PCA	MDS	ProbPCA	GPLVM	KernelPCA
R	0	0.6993	0.6993	0.6873	0.7033	0.7004
	1	0.7183	0.7183	0.7005	0.7183	0.7960
	2	0.7101	0.7101	0.6987	0.7101	0.7989
	3	0.7100	0.7100	0.7158	0.7100	<b>0.7991</b>
	4	0.7189	0.7189	0.7213	0.7189	0.7313
	5	0.7273	0.7273	0.7084	0.7273	0.7388
	6	0.7374	0.7374	0.7380	0.7374	0.7808
	7	0.7369	0.7369	0.7351	0.7369	0.7396
	8	0.7376	0.7376	0.7329	0.7376	0.7406
	9	0.7377	0.7377	0.7397	0.7377	0.7411
	10	0.7379	0.7379	0.7320	0.7379	0.7432
	11	0.7374	0.7374	0.7401	0.7374	0.7465
	12	0.7352	0.7352	0.7346	0.7352	0.7473
	13	0.7381	0.7381	0.7358	0.7381	0.7480
	14	0.7388	0.7388	0.7397	0.7388	0.7497
	15	0.7389	0.7389	0.7365	0.7389	0.7485
	16	0.7397	0.7397	0.7355	0.7397	0.7485
	17	0.7405	0.7405	0.7406	0.7405	0.7882
	18	0.7378	0.7378	0.7338	0.7378	0.7871
	19	<b>0.7407</b>	<b>0.7407</b>	<b>0.7418</b>	<b>0.7407</b>	0.7469
	20	0.7404	0.7404	0.7392	0.7404	0.7465

Tabla 3.21. Resultado para el conjunto de datos balance 1 de 3

		SNE	SymSNE	tSNE	DiffusionMaps	Autoencoder
R	0	0.7596	0.6366	0.7551	0.6618	0.7144
	1	0.7135	0.6461	0.8052	0.7024	0.6678
	2	0.7635	0.6035	0.8025	0.7048	0.6799
	3	0.7644	0.6121	0.8055	0.7083	0.6762
	4	0.7646	0.5951	0.8098	0.7012	0.6008
	5	0.7208	0.6190	0.7637	0.7020	0.6422
	6	0.7745	0.6024	0.8207	0.7215	0.7198
	7	0.7351	<b>0.6726</b>	0.7773	0.7264	0.6961
	8	0.7690	0.5901	0.8264	0.7296	0.7161
	9	0.7790	0.6647	0.7775	0.7299	0.7276
	10	0.7813	0.5502	0.7787	0.7304	0.7189
	11	0.7861	0.6043	0.7837	0.7315	0.7354
	12	<b>0.7907</b>	0.6128	0.8246	0.7305	0.7009
	13	0.7732	0.6302	<b>0.8316</b>	0.7313	0.7380
	14	0.7832	0.6318	0.8253	0.7341	0.7447
	15	0.7798	0.5488	0.8216	0.7361	0.7583
	16	0.7865	0.6062	0.8230	0.7359	<b>0.7729</b>
	17	0.7373	0.5572	0.8239	0.7376	0.7430
	18	0.7808	0.5786	0.8222	0.7414	0.7449
	19	0.7824	0.6381	0.8237	<b>0.7446</b>	0.7096
	20	0.7783	0.6112	0.8195	0.7444	0.7452

Tabla 3.22. Resultados para el conjunto de datos balance 2 de 3

		Laplacian	NPE	MCML	NCA
R	0	0.7241	0.7731	0.6975	<b>0.7624</b>
	1	0.7398	0.6705	0.7390	0.7180
	2	0.7797	0.6095	0.7435	0.7398
	3	0.7799	0.5653	0.7443	0.7483
	4	0.8153	0.6410	0.7505	0.7431
	5	0.7818	0.6435	0.7539	0.7371
	6	0.8211	0.6801	0.7555	0.7480
	7	0.8085	0.5504	0.7567	0.7463
	8	0.7775	0.7007	0.7580	0.7473
	9	0.7380	0.6461	0.7589	0.7390
	10	0.7367	0.6962	0.7575	0.7512
	11	0.7786	0.6625	0.7566	0.7399
	12	0.8270	0.7018	0.7557	0.7473
	13	0.7825	0.6232	0.7547	0.7468
	14	0.7893	0.5988	0.7549	0.7512
	15	0.8234	0.6295	0.7547	0.7512
	16	0.8204	0.6277	0.7549	0.7509
	17	0.8246	0.6961	0.7553	0.7515
	18	0.8214	<b>0.7166</b>	0.7555	0.7470
	19	0.8261	0.6864	0.7555	0.7512
	20	<b>0.8271</b>	0.7036	<b>0.8314</b>	0.7436

Tabla 3.23 Resultados para el conjunto de datos balance 3 de 3

Tras mostrar todas las ejecuciones de cada una de las técnicas y marcar el resultado que ha proporcionado un mejor valor en nuestra función de coste, se procede a ordenar las salidas de las distintas técnicas de reducción de dimensionalidad de mejor a peor resultado mostrando a su vez para qué ejecución se han dado.

### 3.8.5 Resultados de salida clasificados

Iris			Wine		
Método	Salida	Factor de x	Método	Salida	Factor de x
SNE	0.8176	1.5	tSNE	0.8336	2.0
LDA	0.8154	1.6/1.9	MCML	0.8325	1.2
tSNE	0.8146	0.9	SNE	0.8298	1.4
MCML	0.8025	0.1	LPP	0.8283	1.0
KernelPCA	0.8023	1.2	Autoencoder	0.8276	2.0
LPP	0.7698	1.6	LDA	0.8254	0.5
Laplacian	0.7541	1.5	NCA	0.8235	1.6
PCA	0.7530	1.6	Laplacian	0.8232	2.0
MDS	0.7530	1.6	Diff.Maps	0.8229	2.0
GPLVM	0.7530	1.6	Sammon	0.8224	1.5/2.0
NCA	0.7506	0.8	ProbPCA	0.8224	2.0
ProbPCA	0.7435	1.9	KernelPCA	0.8180	1.8/1.9/2.0
Autoencoder	0.7323	0.8	PCA	0.8163	1.4/1.5
Diff.Maps	0.7171	1.7	MDS	0.8163	1.4/1.5
SymSNE	0.6540	0.9	GPLVM	0.8163	1.4/1.5
			NPE	0.8053	1.1
			SymSNE	0.7167	1.9

Tabla 3.24 Resultados ordenados de mayor a menor eficiencia para cada conjunto de datos 1 de 2



Car			Balance		
Método	Salida	Factor de x	Método	Salida	Factor de x
NCA	0.8350	0.7	tSNE	0.8316	1.3
KernelPCA	0.8310	2.0	MCML	0.8314	2.0
LDA	0.8306	1.0	Laplacian	0.8271	2.0
MCML	0.8277	0.7	KernelPCA	0.7991	0.3
SNE	0.8247	1.5	SNE	0.7907	1.2
tSNE	0.8242	1.8	Autoencoder	0.7729	1.6
Diff. Maps	0.8232	0.5	NCA	0.7626	0.0
Laplacian	0.8129	0.6	Diff. Maps	0.7446	1.9
GPLVM	0.8087	0.0	ProbPCA	0.7418	1.9
MDS	0.8081	0.0	GPLVM	0.7407	1.9
Autoencoder	0.8043	1.9	PCA	0.7407	1.9
ProbPCA	0.8017	1.5	MDS	0.7407	1.9
PCA	0.7858	1.5	NPE	0.7166	1.8
			SymSNE	0.6726	0.7

Tabla 3.25 Resultados ordenados de mayor a menor eficiencia para cada conjunto de datos 2 de 2

### 3.9 Representación gráfica de resultados

Tras la obtención de los mejores resultados para cada técnica de reducción de dimensionalidad llega el momento de representar dicha ejecución por pantalla. Para poder ver así si existe por un lado una correspondencia entre el resultado numérico y la representación visual. Y por otro comparar las representaciones de las distintas técnicas de reducción de dimensionalidad.

Para la creación de la frontera se hará uso otra vez de la herramienta de SVM. Esto es debido a que para la realización de dicha frontera nos basaremos en la representación de gran cantidad de puntos que serán predichos por el modelo de SVM. Ya se presentó el código de SVM para la predicción de datos, este era de la siguiente forma `[predict_label,accuracy,prob_values]=svmpredict( y, x, model)`.

En este apartado lo que interesa es `predict_label`, es decir la etiqueta que el predictor ha elegido para cada punto. Entonces, para poder realizar la frontera se seguirán los siguiente pasos.

- Tras realizar la ejecución para la cual se ha obtenido el mejor valor de la función de coste procederemos a realizar la transformación del espacio  $N$  dimensional a dos dimensiones.
- Posteriormente se crearan dos variables `X_axis` e `Y_axis` que serán las encargadas de representar los dos ejes. Dichas variables se calculan de la siguiente forma. Se toma la matriz resultado de la reducción de dimensionalidad y para cada eje se recorre desde el valor mínimo al máximo con saltos elegidos por el usuario. De esta forma conseguimos representar puntos intermedios entre los puntos en los que se representan nuestros datos
- Después creamos un vector que guarde todos los puntos posibles, es decir recorreremos `X_axis` sobre `Y_axis` para obtener todos los puntos posibles, esto da un vector con una longitud de `X_axis * Y_axis`. Este vector será la entrada al predictor como si fueran datos a predecir
- Las etiquetas de dichos datos se inicializan a 1, ya que no interesa la efectividad, tan solo la predicción que hará de dichos puntos SVM.
- Finalmente se procede a pintar las etiquetas predichas, de tal forma que en nuestro espacio aparecerán las fronteras de decisión, y sobre ellas podremos pintar ahora los conjuntos de entrenamiento y test.

El resultado final obtenido se puede ver en los siguientes ejemplos donde se van a presentar las tres mejores salidas y las dos peores para cada uno de los conjuntos de datos.

### 3.9.1 Representaciones del conjunto Iris

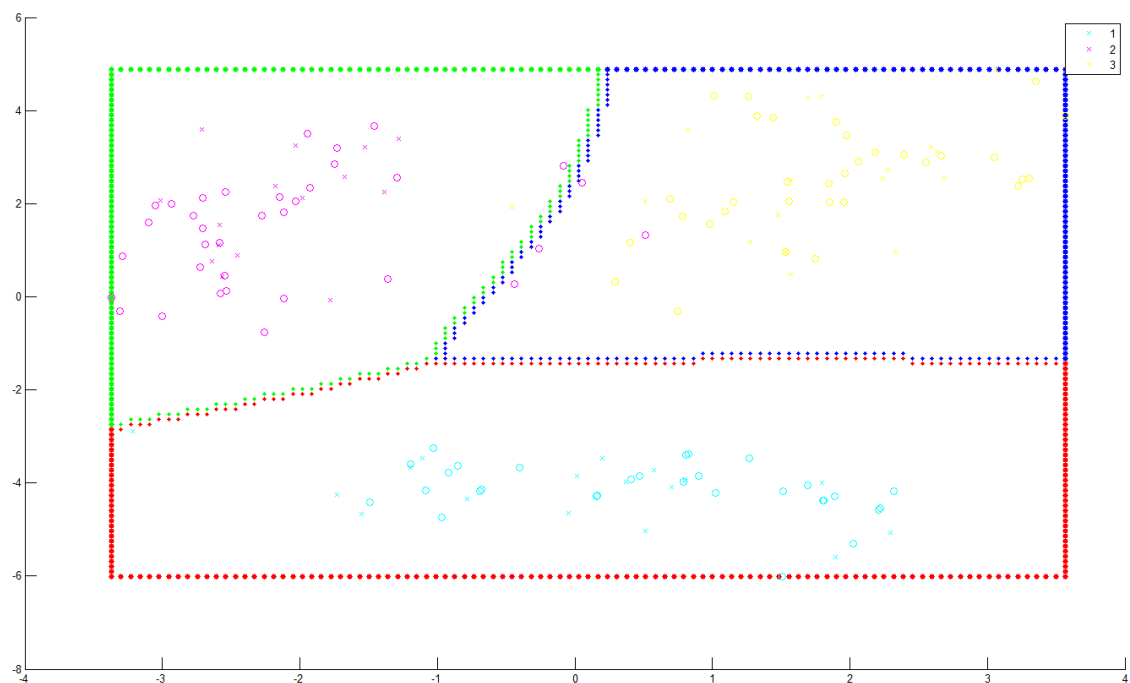


Figura 3.7. Imagen del conjunto iris sobre el método de SNE

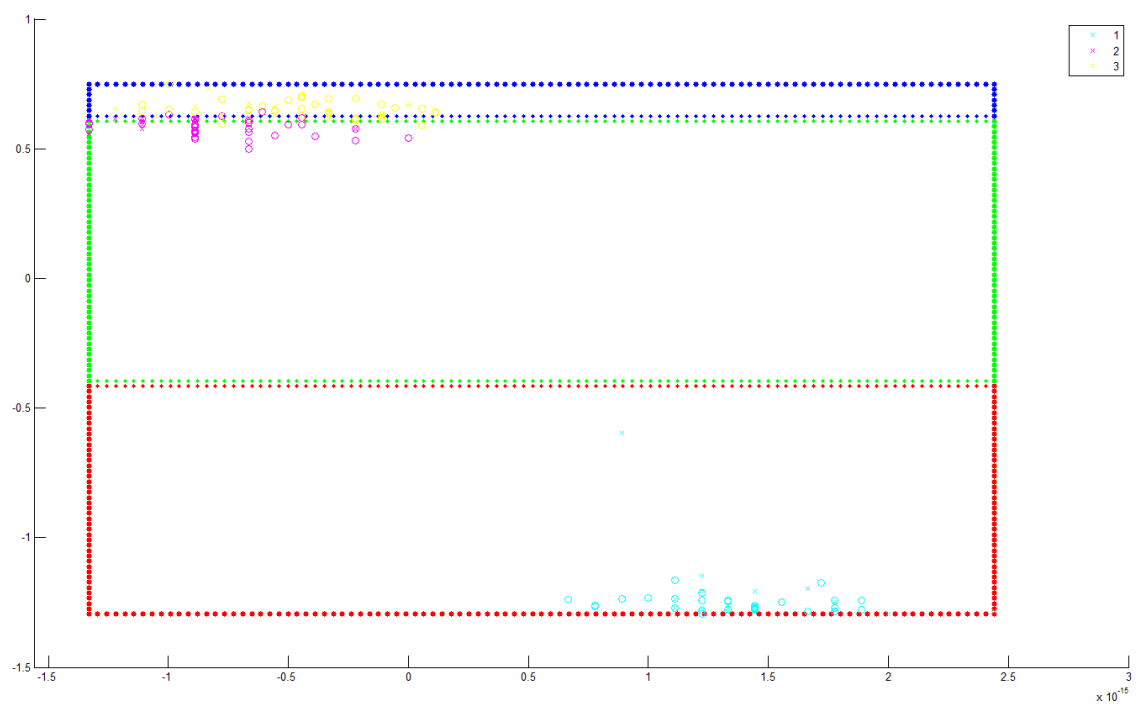


Figura 3.8. Imagen del conjunto iris sobre el método de LDA

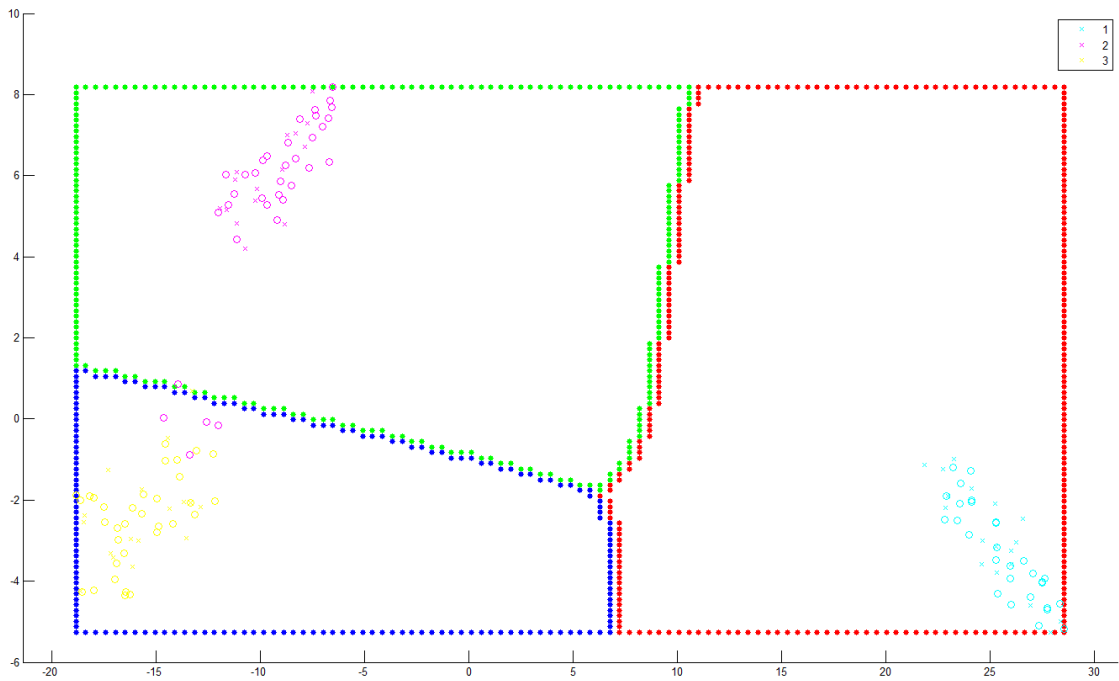


Figura 3.9. Imagen del conjunto iris sobre el método de tSNE

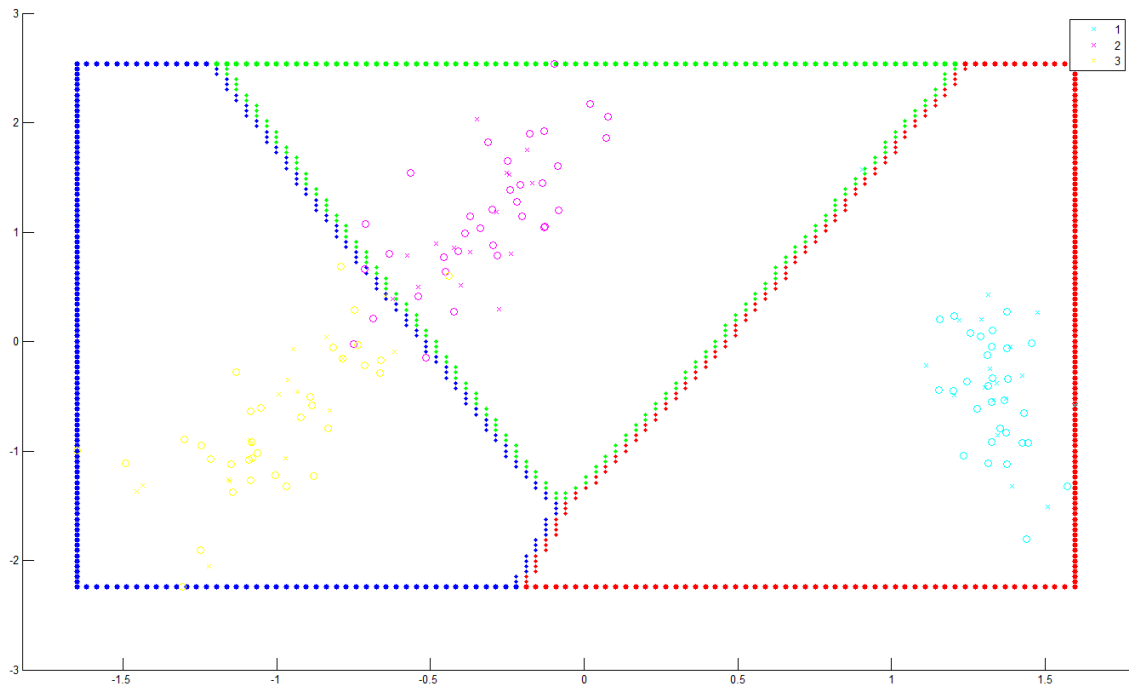


Figura 3.10. Imagen del conjunto iris sobre el método de Diffusion Maps

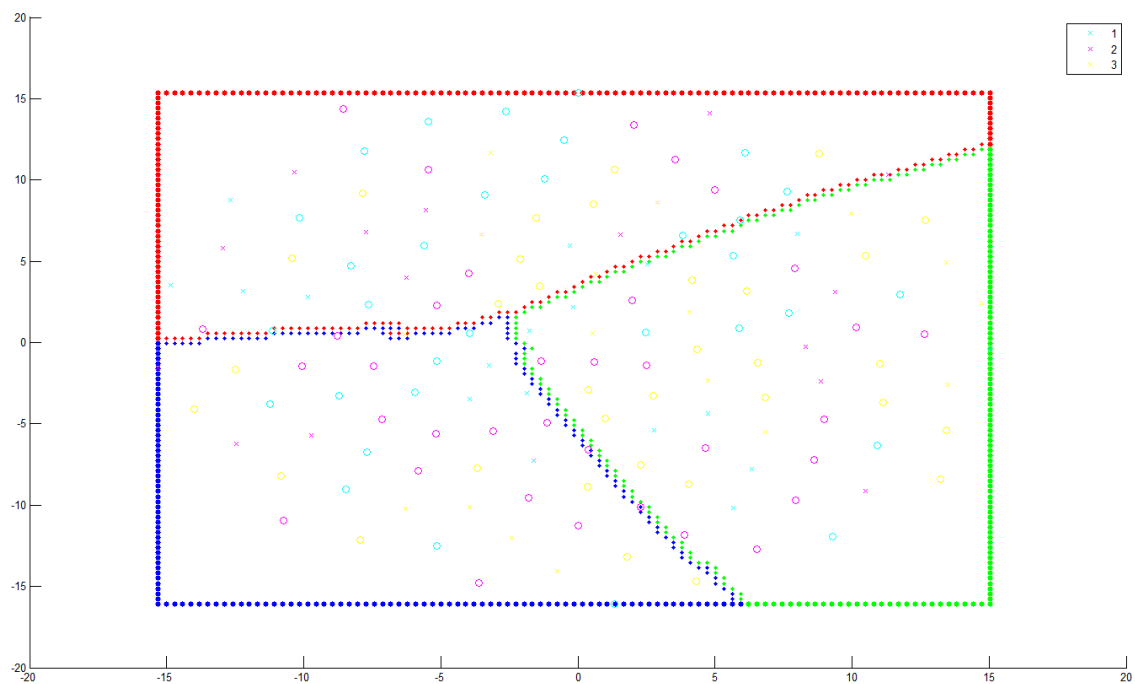


Figura 3.11. Imagen del conjunto iris sobre el método de SymSNE

### 3.9.2 Representaciones del conjunto Wine

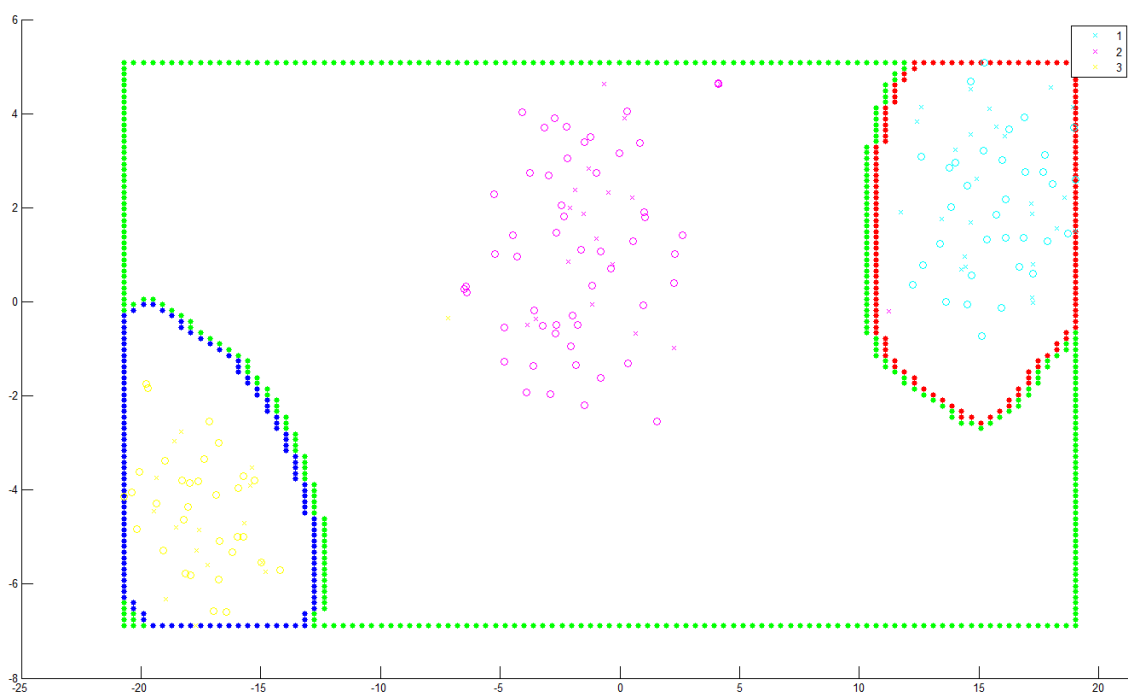


Figura 3.12. Imagen del conjunto wine sobre el método de tSNE

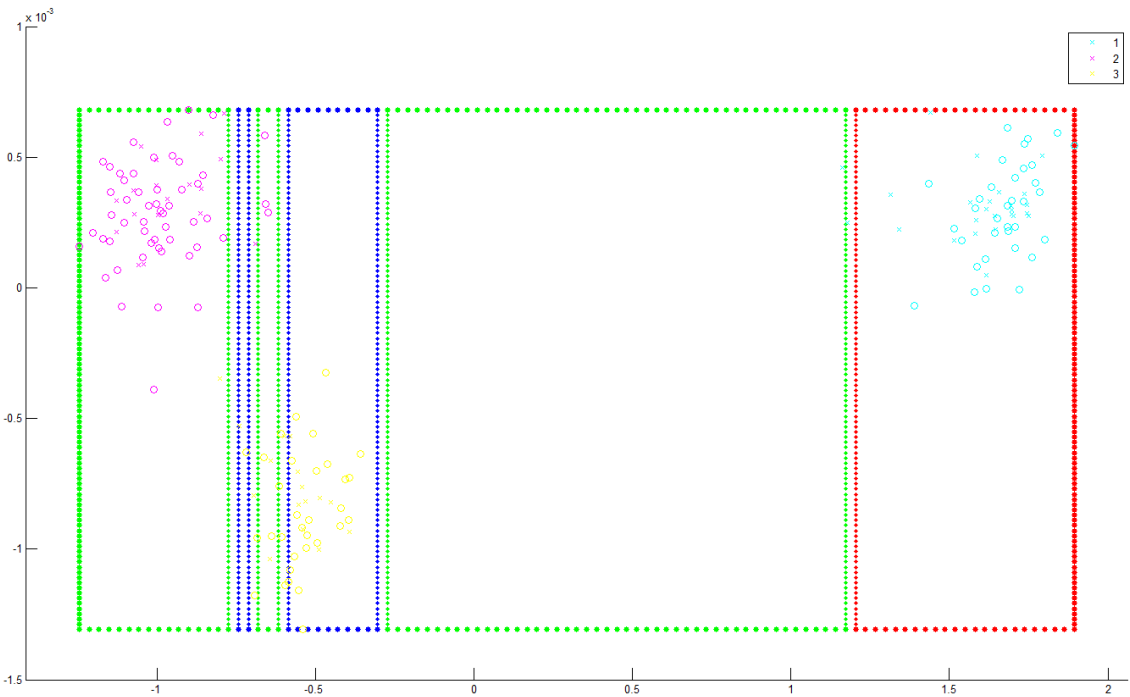


Figura 3.13. Imagen del conjunto wine sobre el método de MCML

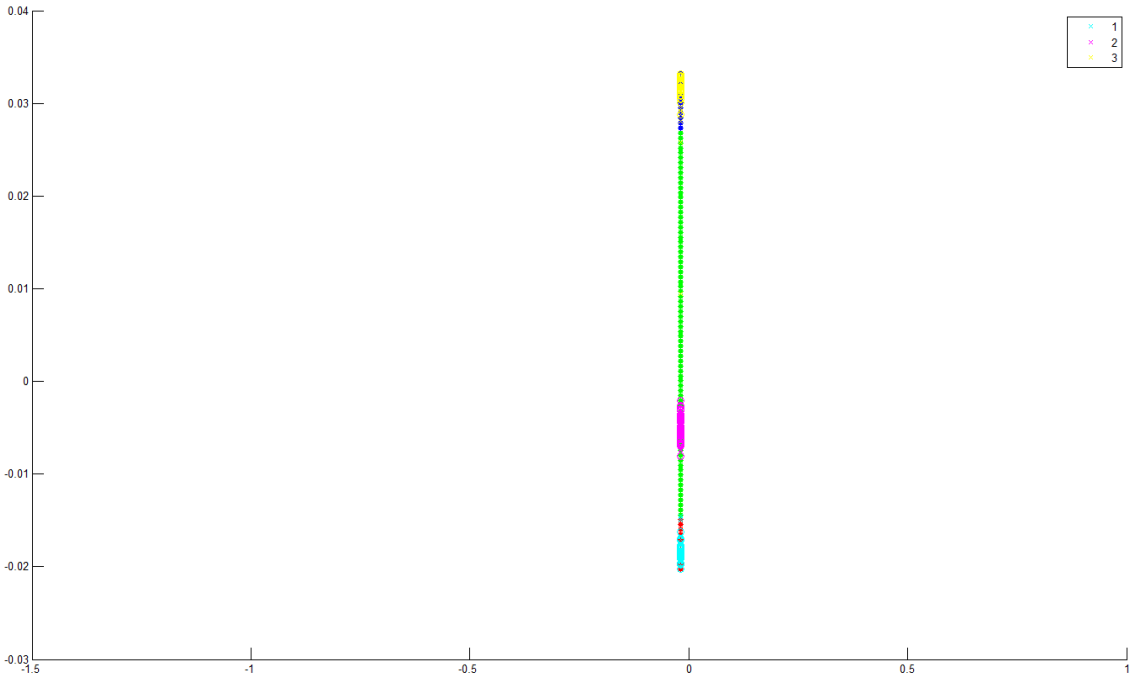


Figura 3.14. Imagen del conjunto wine sobre el método de LPP

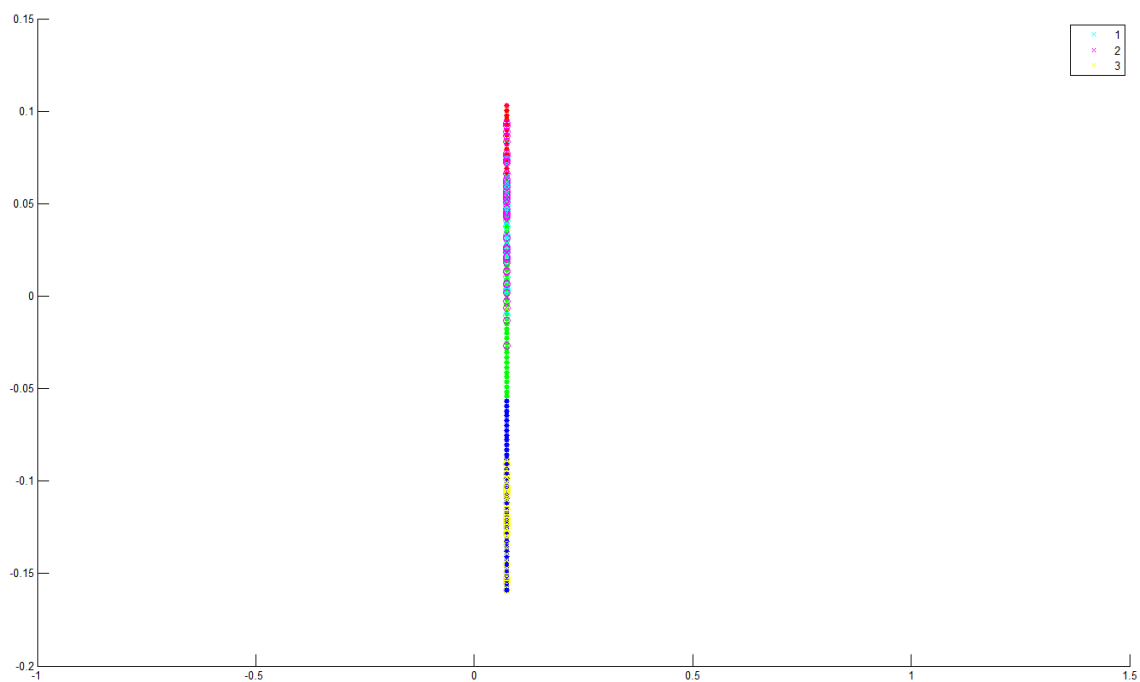


Figura 3.15. Imagen del conjunto wine sobre el método de NPE

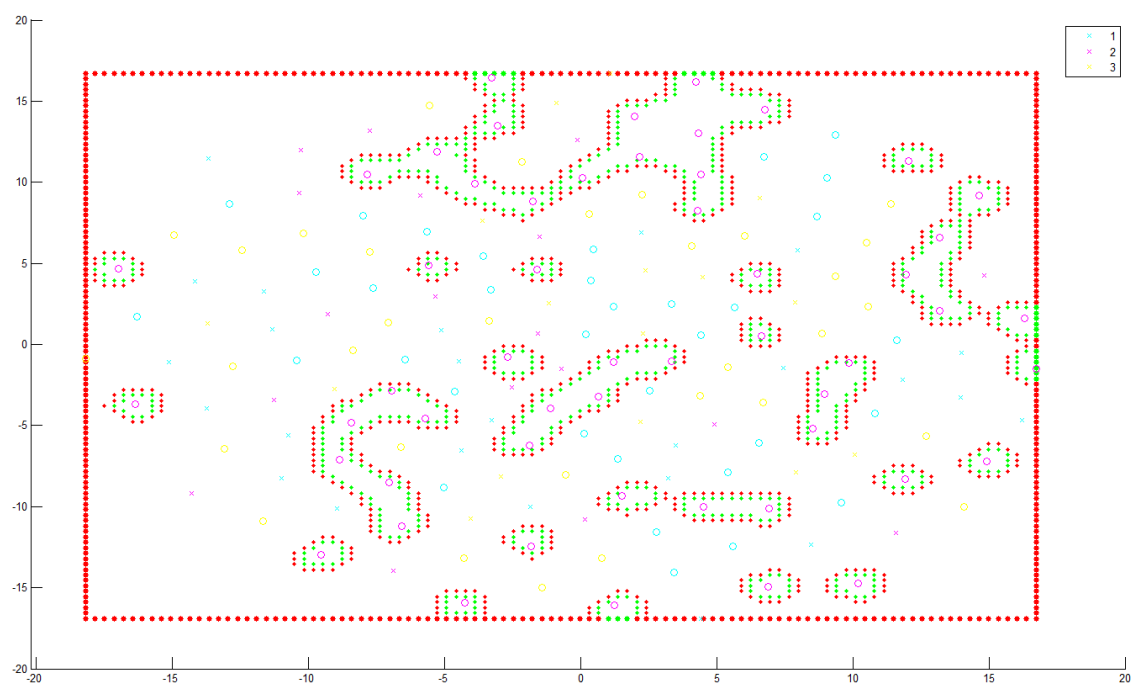


Figura 3.16. Imagen del conjunto wine sobre el método de SymSNE

3.9.3 Representaciones del conjunto Balance

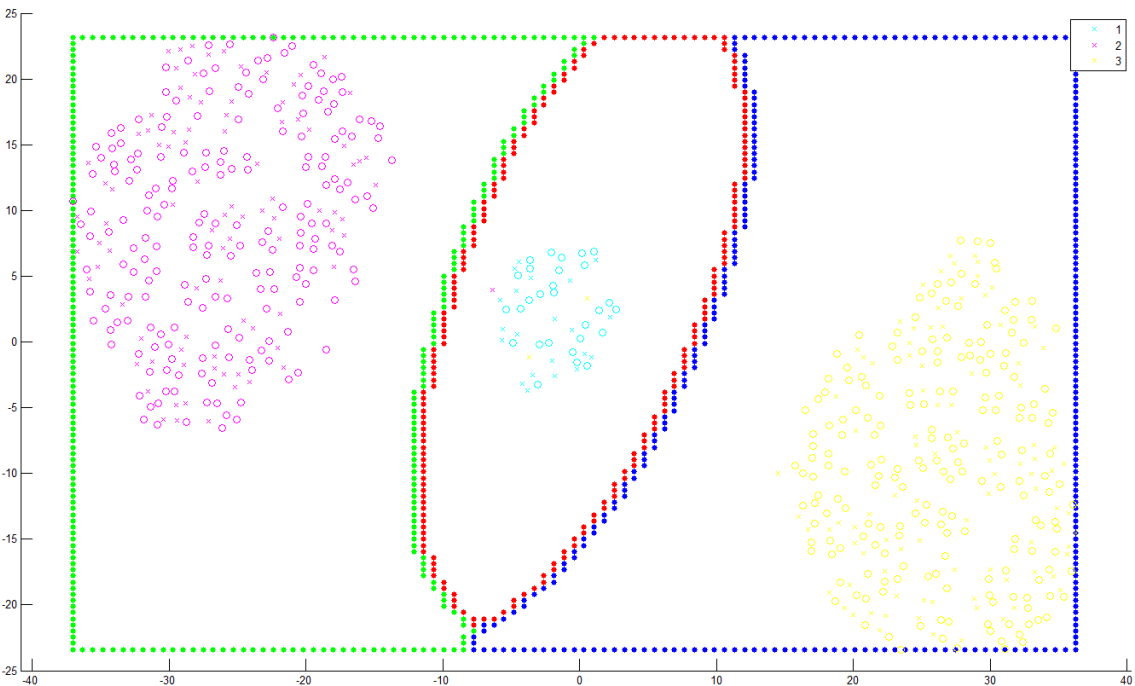


Figura 3.17. Imagen del conjunto balance sobre el método de tSNE

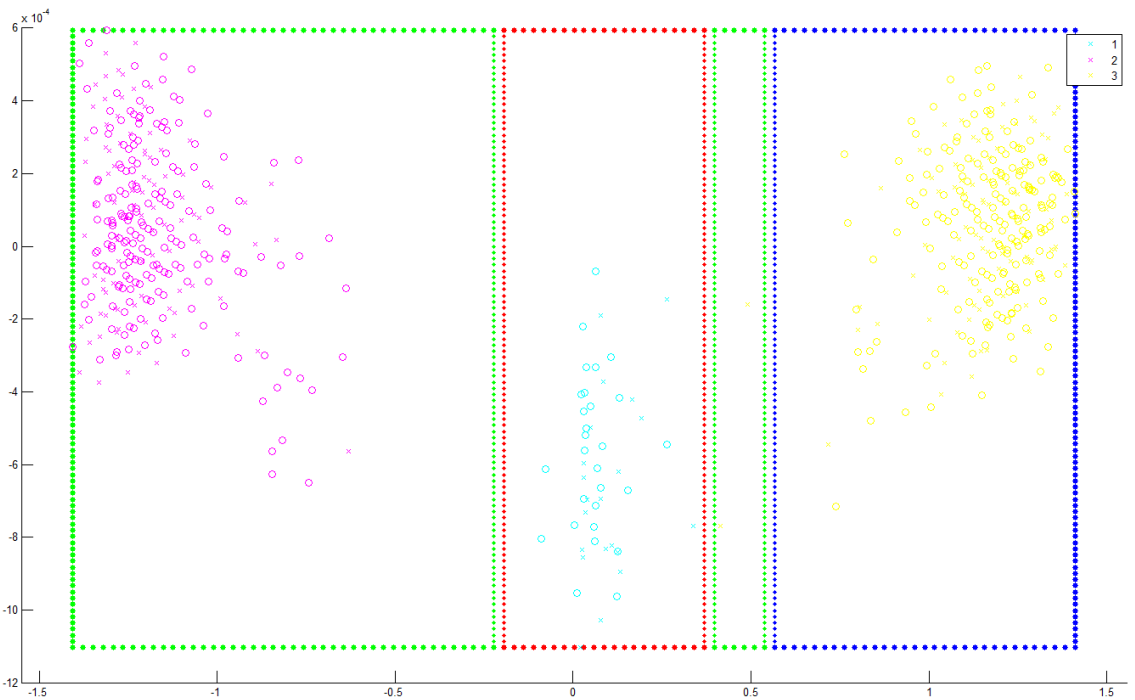


Figura 3.18. Imagen del conjunto balance sobre el método de MCML



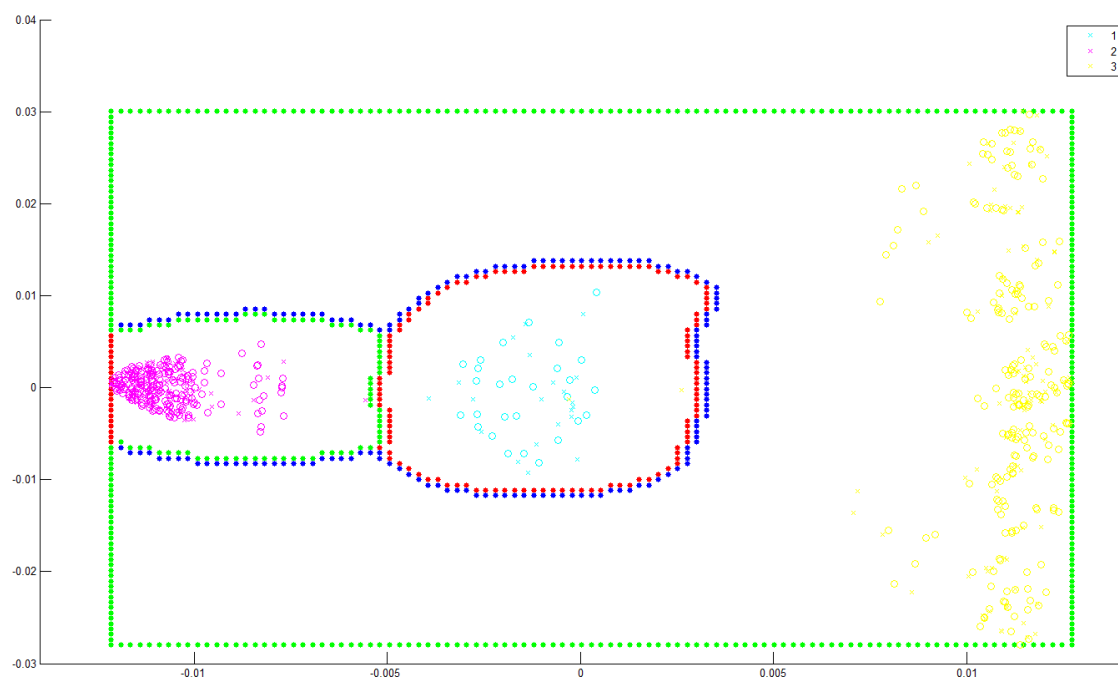


Figura 3.19. Imagen del conjunto balance sobre el método de Laplacian

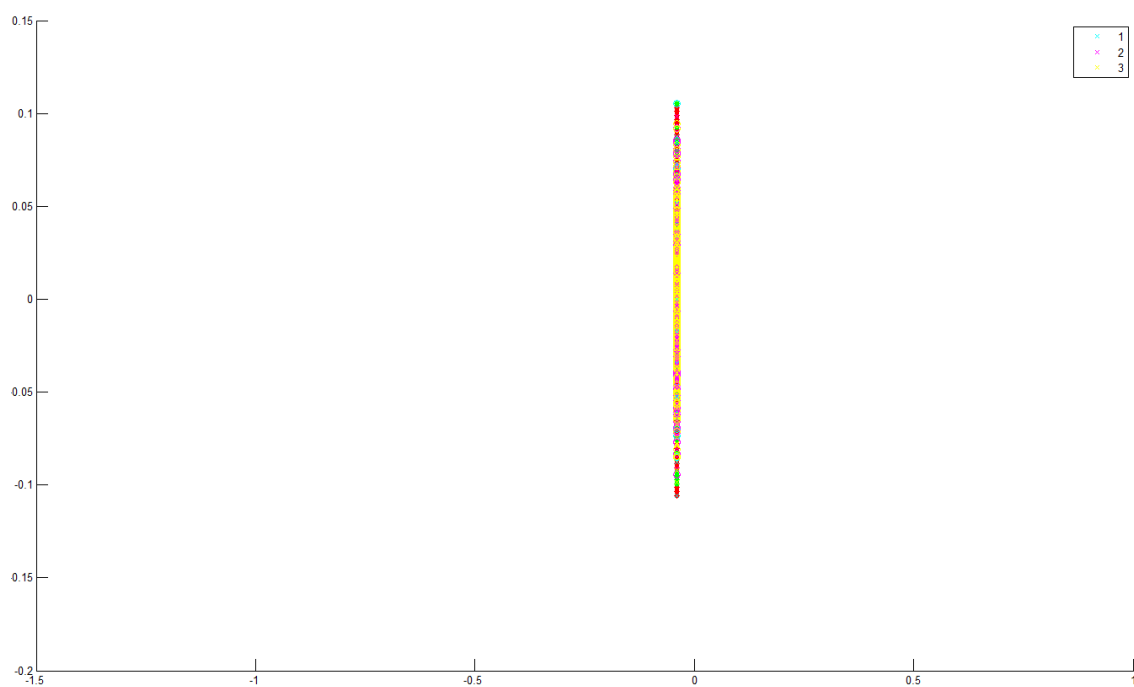


Figura 3.20. Imagen del conjunto balance sobre el método de NPE

## Visualización bidimensional de problemas de clasificación en alta dimensión

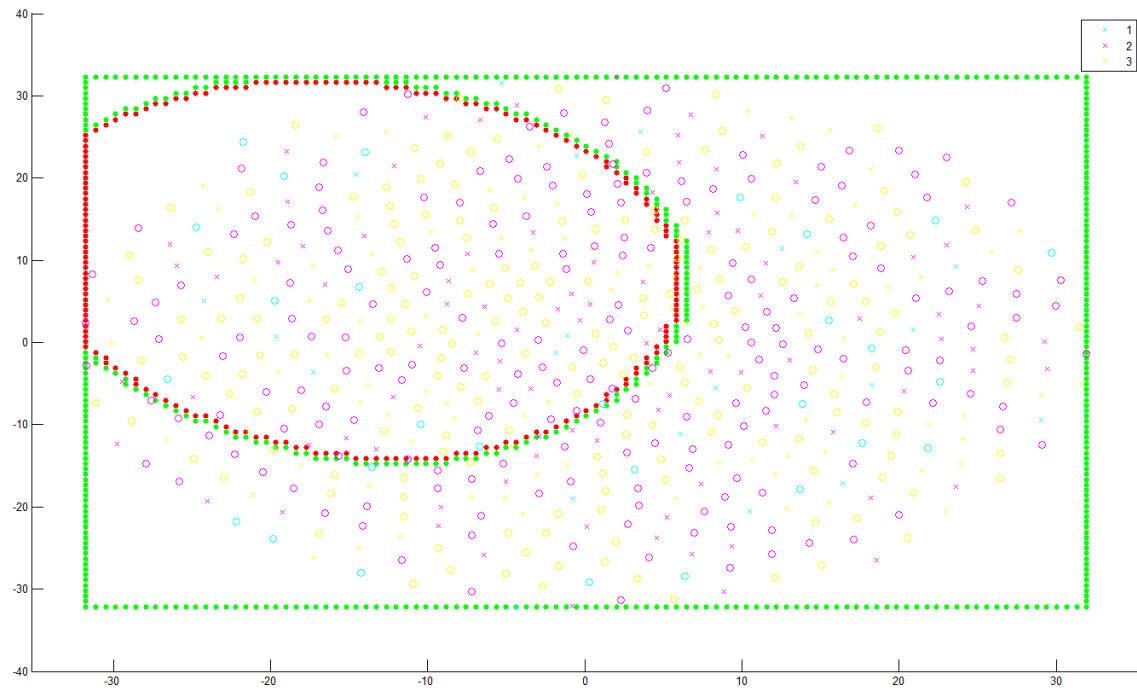


Figura 3.21. Imagen del conjunto balance sobre el método de SymSNE

### 3.9.4 Representaciones del conjunto Car

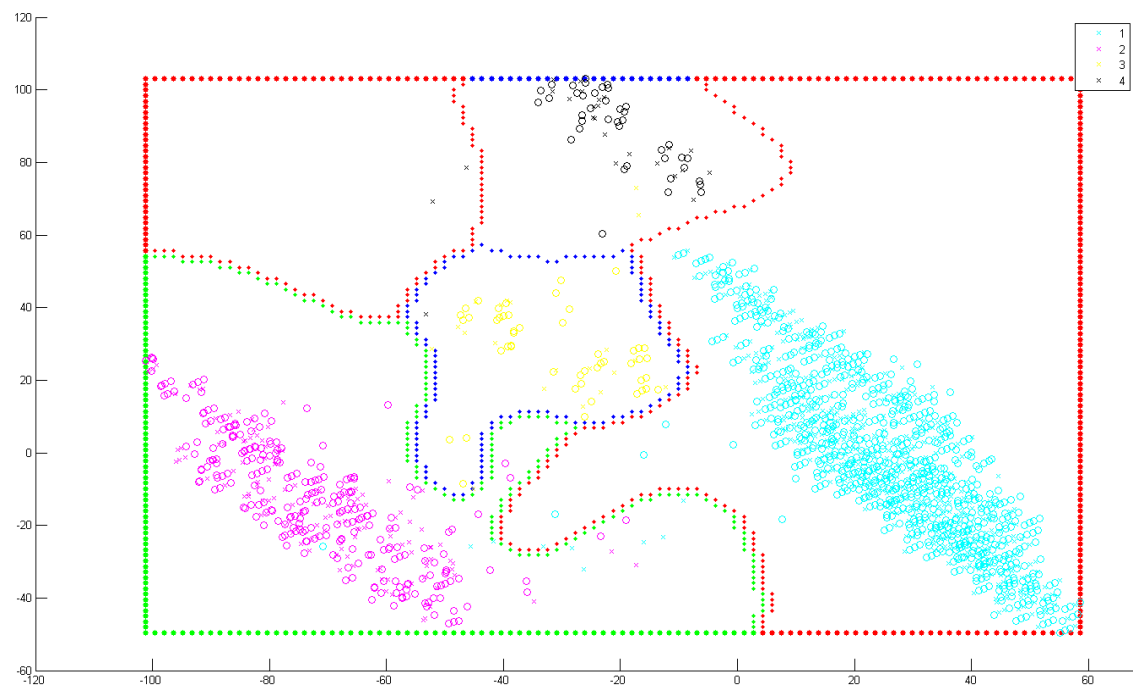


Figura 3.22. Imagen del conjunto car sobre el método de NCA

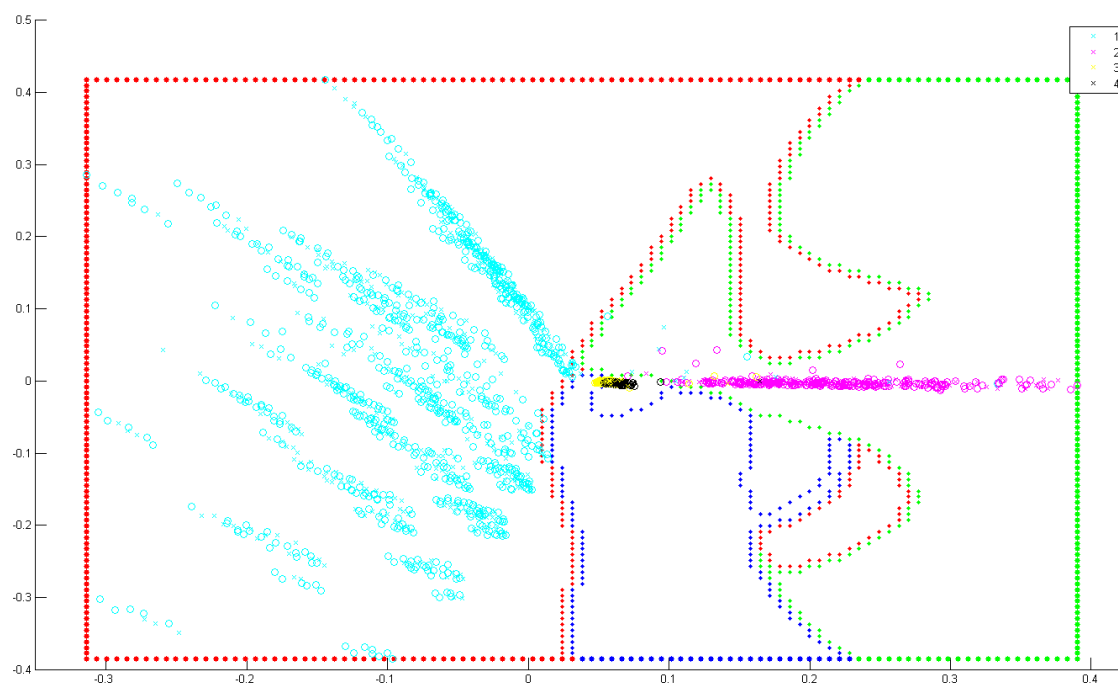


Figura 3.23. Imagen del conjunto car sobre el método de KernelPCA

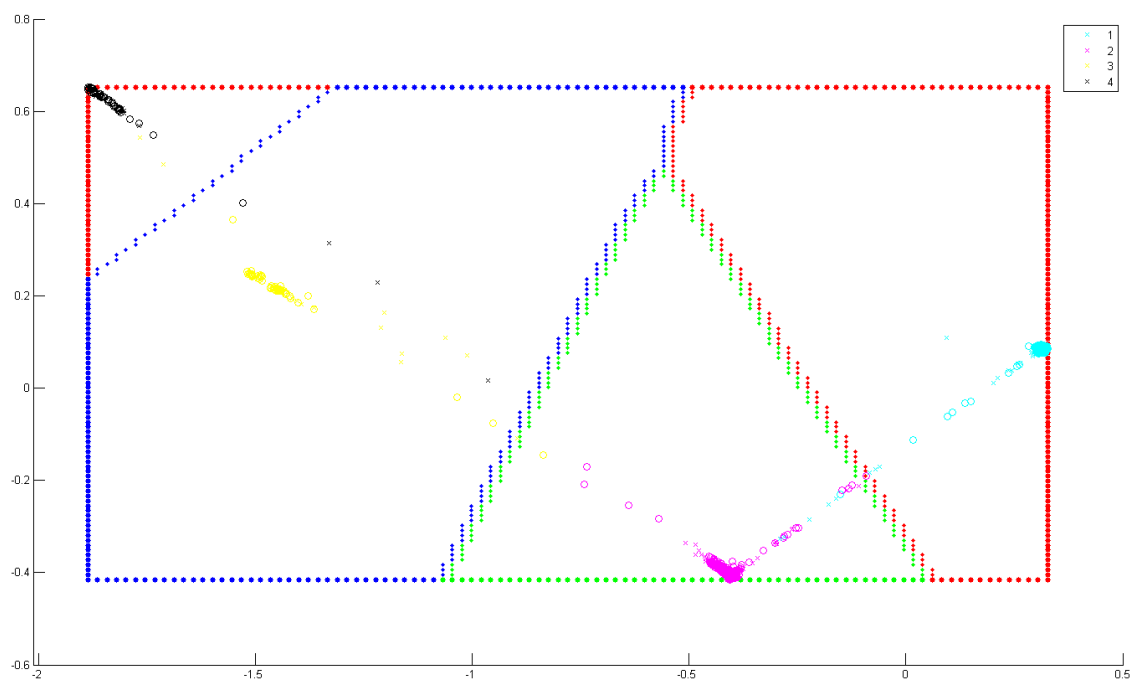


Figura 3.24. Imagen del conjunto car sobre el método de LDA

## Visualización bidimensional de problemas de clasificación en alta dimensión

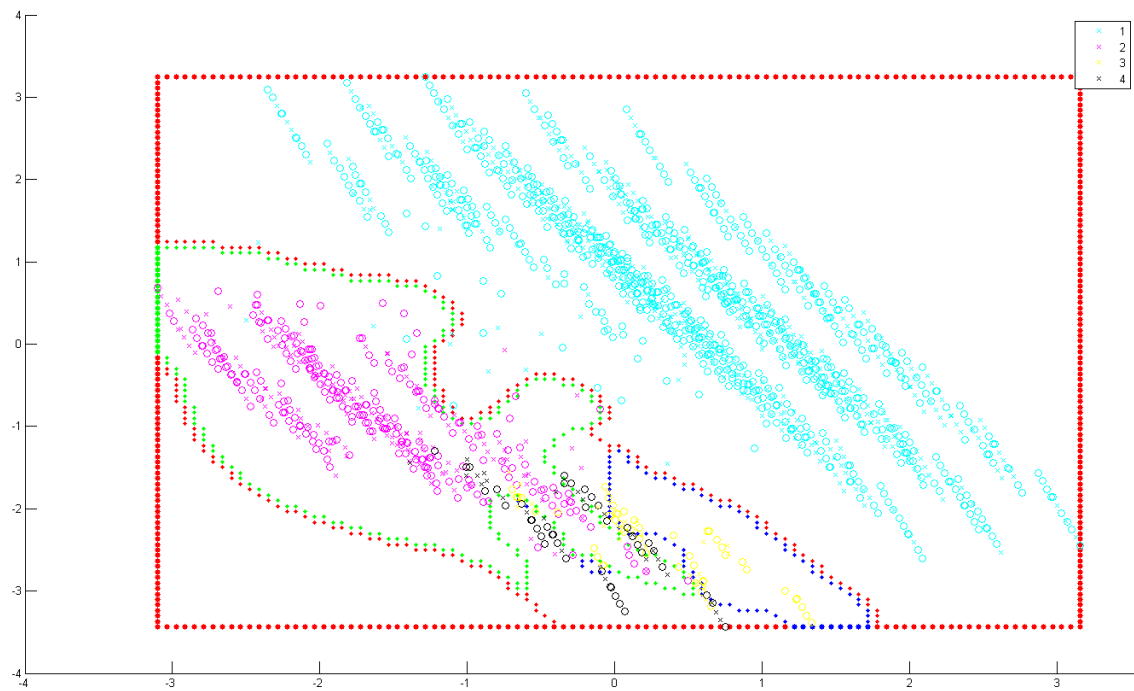


Figura 3.25. Imagen del conjunto car sobre el método de ProbPCA

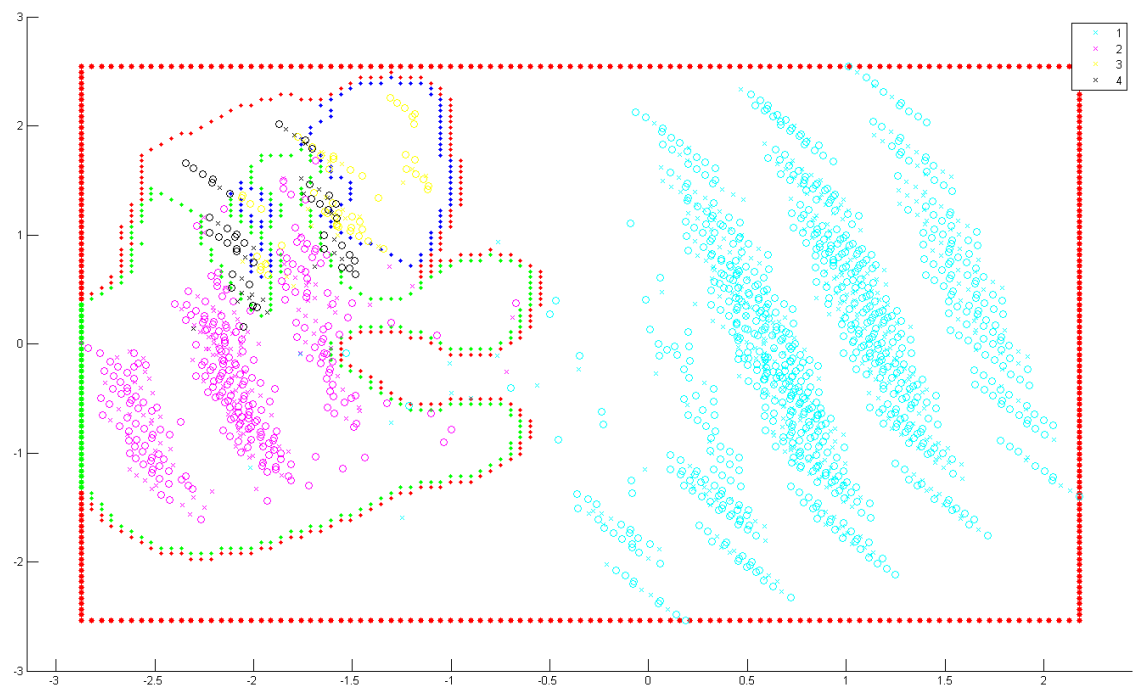


Figura 3.26. Imagen del conjunto car sobre el método de PCA

## Capítulo 4: Conclusiones y trabajos futuros

### 4.1 Conclusiones

Las conclusiones de este proyecto las podemos enlazar con los objetivos de la siguiente forma.

- Sub-objetivo 1: De los 59 resultados obtenidos, en tan solo tres casos han sido mejores aquellos que contaban con las características iniciales, en todos los demás casos los mejores resultados aparecían cuando existía información adicional. Se puede decir que en los casos estudiados se encuentra una predisposición a mostrar unos mejores resultados cuando es añadida la salida blanda del clasificador.
- Sub-objetivo 2: Se ha conseguido crear una función de coste basada en tres parámetros los cuales permiten realizar una comparación de forma objetiva de las distintas técnicas de reducción de dimensionalidad.
- Sub-objetivo 3: Se ha podido realizar una clasificación de las distintas técnicas de reducción de dimensionalidad pudiendo así ordenar de mejor a peor las distintas visualizaciones. Se puede observar una tendencia hacia el teorema “No free lunch”, en los ejemplos estudiados no encontramos ninguna predominancia de ningún método. Vemos como algunas técnicas de reducción suelen posicionarse en los primeros puestos de clasificación como NCA, tSNE, MCML y SNE y otras que en cambio en los ejemplos estudiados tienden a situarse en la parte baja de la clasificación como son MDS, PCA y SymSNE.
- Objetivo global: Gracias a la realización y a los buenos resultados obtenidos en los distintos sub-objetivos se ha podido realizar el objetivo global. Se han conseguido visualizar los datos de espacios  $N$  dimensionales en entornos bidimensionales y gracias a los resultados se ha podido realizar una clasificación de los mismos.

Con estos resultados se ha conseguido avanzar un paso más en las técnicas de reducción de dimensionalidad y en la visualización de las mismas pudiendo hacer la comparación entre ellas y poder cuantificar la diferencia entre unos métodos y otros, permitiendo así la clasificación realizada.

## 4.2 Trabajos futuros

Si se tuviera que hablar sobre posibles trabajos futuros hay que tener en cuenta que hay distintas ramas hacia donde se podría avanzar.

- Estudio sobre los resultados: En algunas de las representaciones, LPP y NPE, se obtienen unas representaciones que no permiten la buena visualización de los datos, ya que estos se encuentran agrupados en una pequeña franja. El objetivo de este proyecto es que la función de coste sea una medida sobre lo buena o mala que es la visualización del conjunto. Este caso debería estudiarse en profundidad ya que se imposibilita la visualización de la forma en que se representan los datos.
- Métodos de reducción de dimensionalidad: Se podrían ampliar los métodos de reducción de dimensionalidad, buscar los errores de implementación de las técnicas que no han sido posibles estudiar y presentar los resultados para poder hacer la comparativa con todas las técnicas de la librería. De la misma forma se ha utilizado durante la realización del proyecto los valores por defecto de las técnicas de reducción de dimensionalidad y con un estudio más exhaustivo se podrían ajustar los parámetros al conjunto de datos para obtener los mejores resultados posibles.
- Función de coste del proyecto: Otro camino posible para la mejora de este trabajo sería realizar un estudio sobre los pesos de los parámetros y sobre el peso de cada uno de los conjuntos de datos en la variable de salida de forma que pueda optimizar la función de coste desarrollada. En este trabajo se ha realizado de la siguiente forma. Se ha dado el mismo peso al conjunto de entrenamiento como al conjunto de test, aun siendo este más pequeño que el anterior. Surgiría la duda de si debería tener más peso el de test, ya que siempre existirán valores futuros y son los que van a usarse sobre el modelo y

queremos el mejor resultado posible, o por si el contrario deberíamos dar más importancia al de entrenamiento ya que es la base para posteriores ejecuciones, quitándole así importancia al de test. También habría que estudiar si los tres parámetros que se han utilizado para la realización del método, eficiencia, Kendall- $\tau$  y comparación de errores, deberían tener el mismo peso sobre el total. Puede que la clasificación mostrase unos resultados distintos si variásemos los pesos de estas características sobre el conjunto total. Otra forma de mejorar la función de coste podría ser inventando nuevos parámetros de medición.

- Ayuda de expertos: Otra rama de avance en este proyecto sería que una base de datos fuese evaluada por expertos de la materia, de esta forma al representar las visualizaciones de los distintos datos, podrían ayudar a entender el por qué de esas visualizaciones y sobre todo analizar si se ha conseguido una buena distribución de los datos respecto al espacio original base de N dimensiones o si por el contrario ha realizado una transformación bidimensional alterando la ordenación de los datos.
- Modificación de función de coste de las técnicas de reducción de dimensionalidad existentes: Si a las técnicas ya descritas se las añadiese la información que hemos obtenido en la función de coste, es decir, ampliando su función de coste con los parámetros estudiados en este proyecto, se podría pensar que algunas de ellas pudieran devolver unos mejores resultados de salida. Esto llevaría a una mejor representación del conjunto y a una menor transformación del espacio N dimensional.

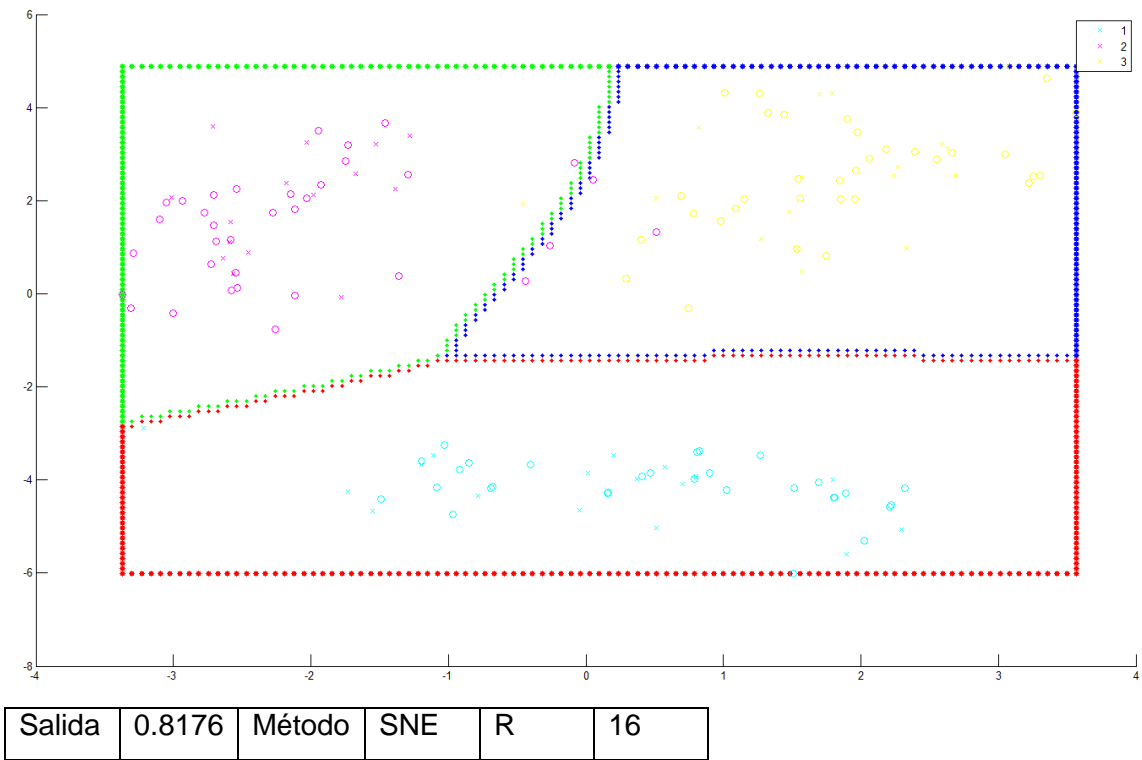
# Capítulo 5: Anexos

## 5.1 Representaciones

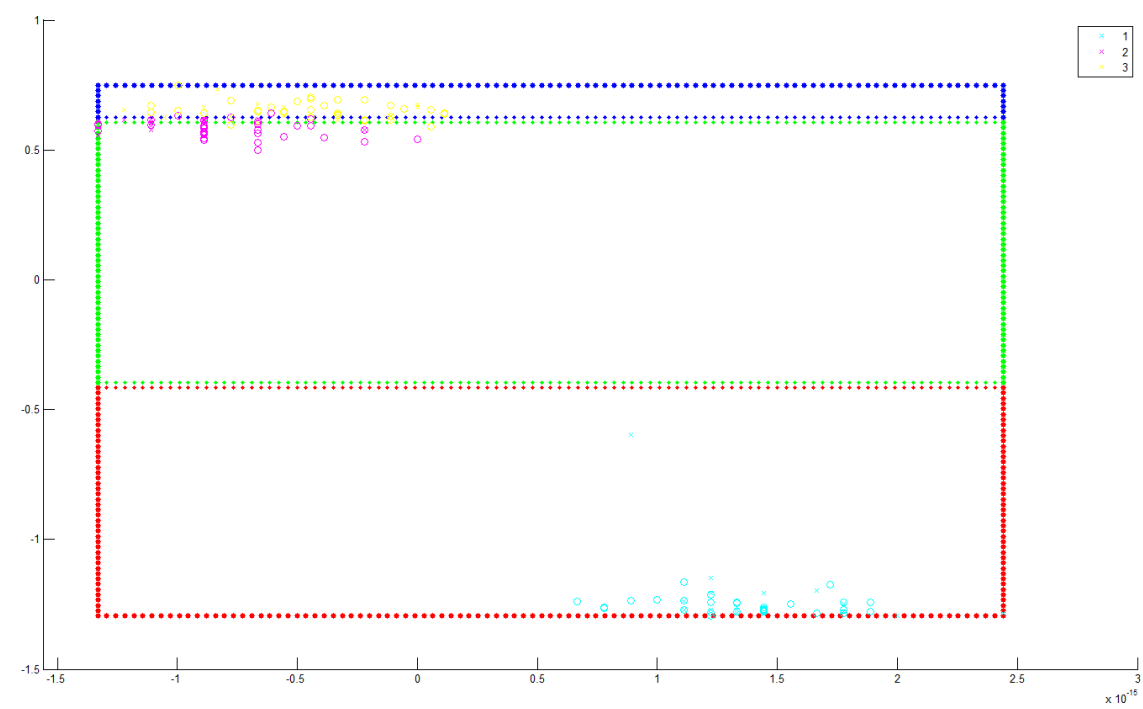
En este anexo se muestran todas las representaciones obtenidas para cada uno de los conjuntos de datos sobre cada técnica en la mejor ejecución de R.

Dichas representaciones están ordenadas de mejor a peor y cada una tiene la información sobre su pie.

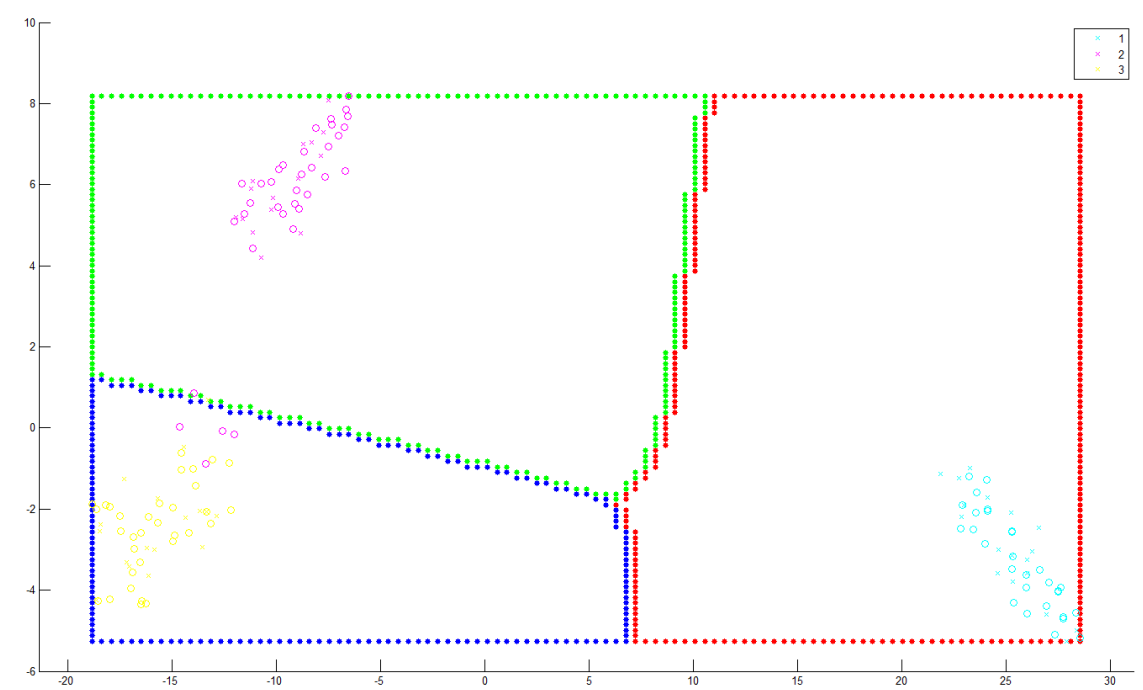
### 5.1.1 Conjunto iris





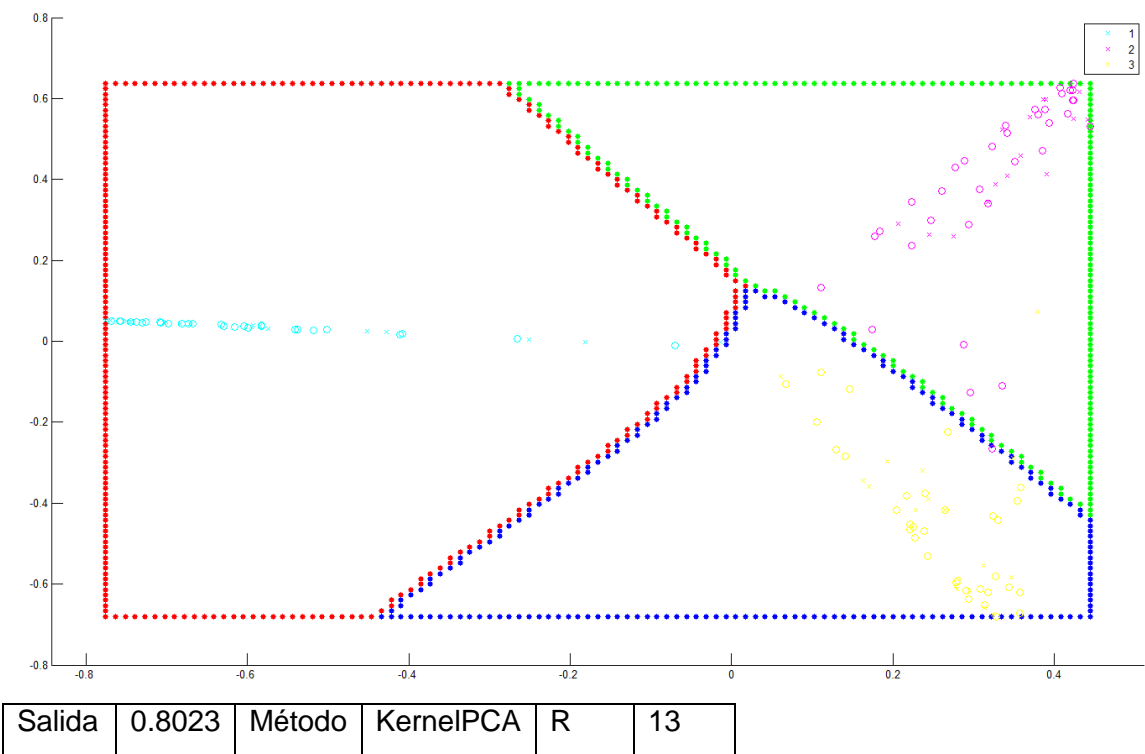
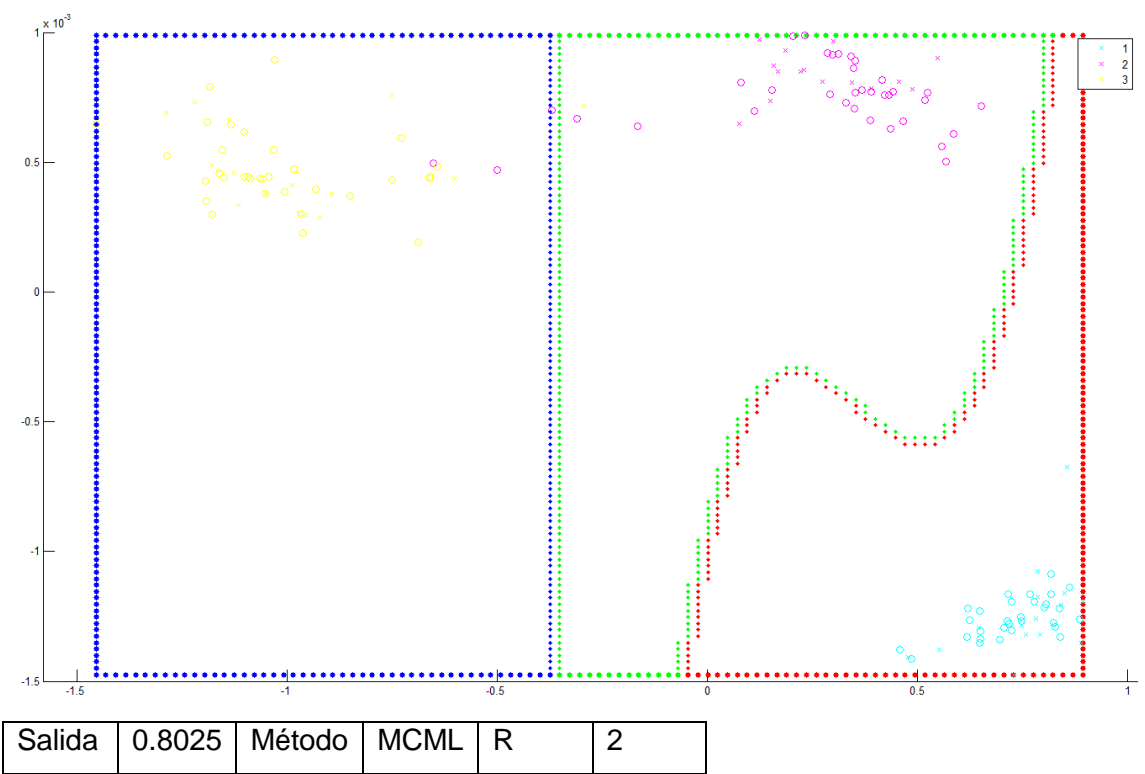


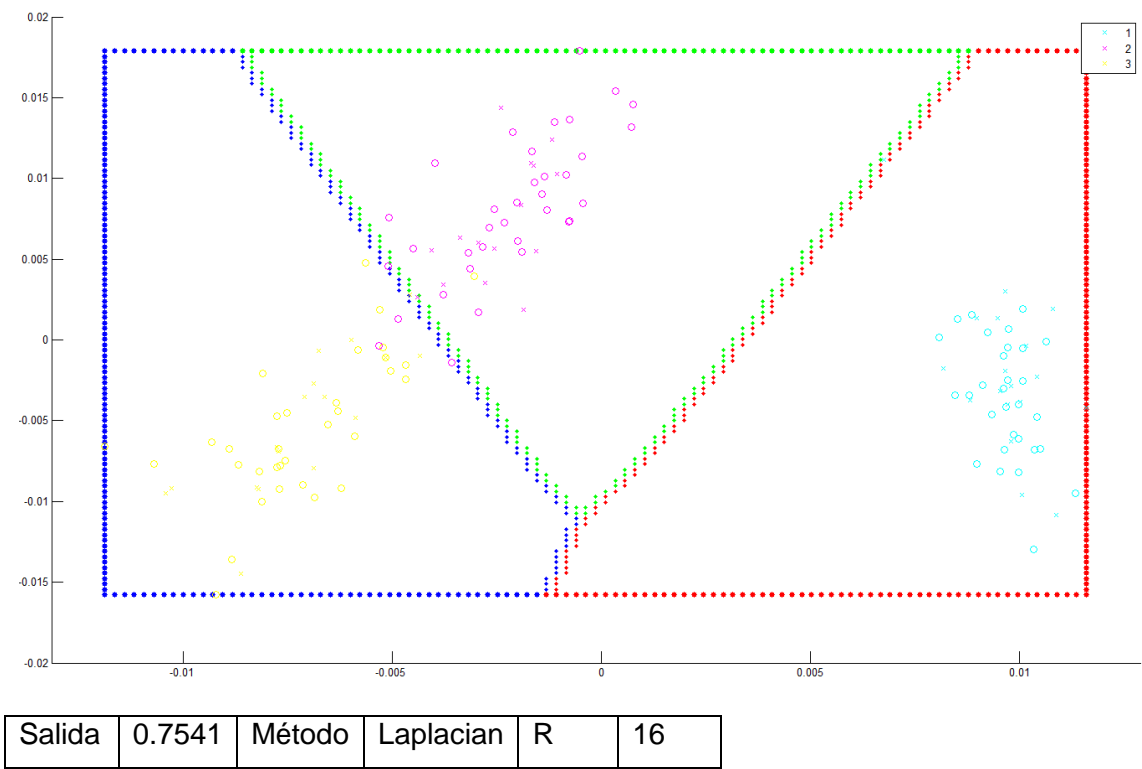
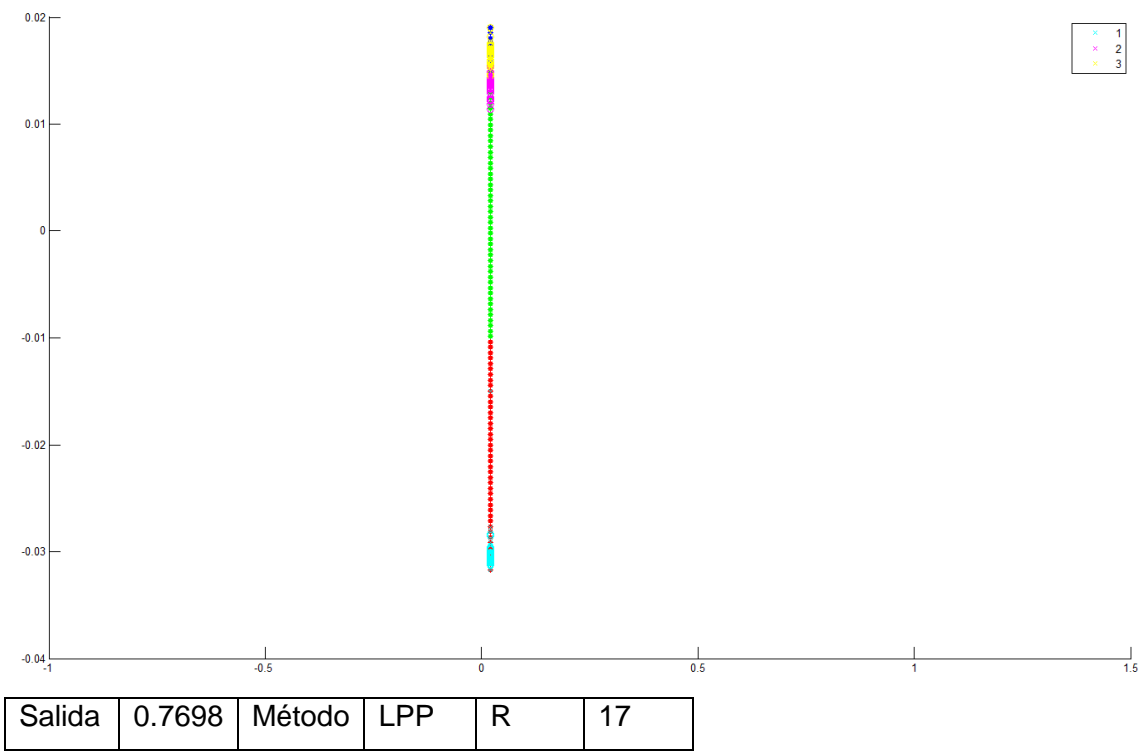
Salida	0.8154	Método	LDA	R	17
--------	--------	--------	-----	---	----



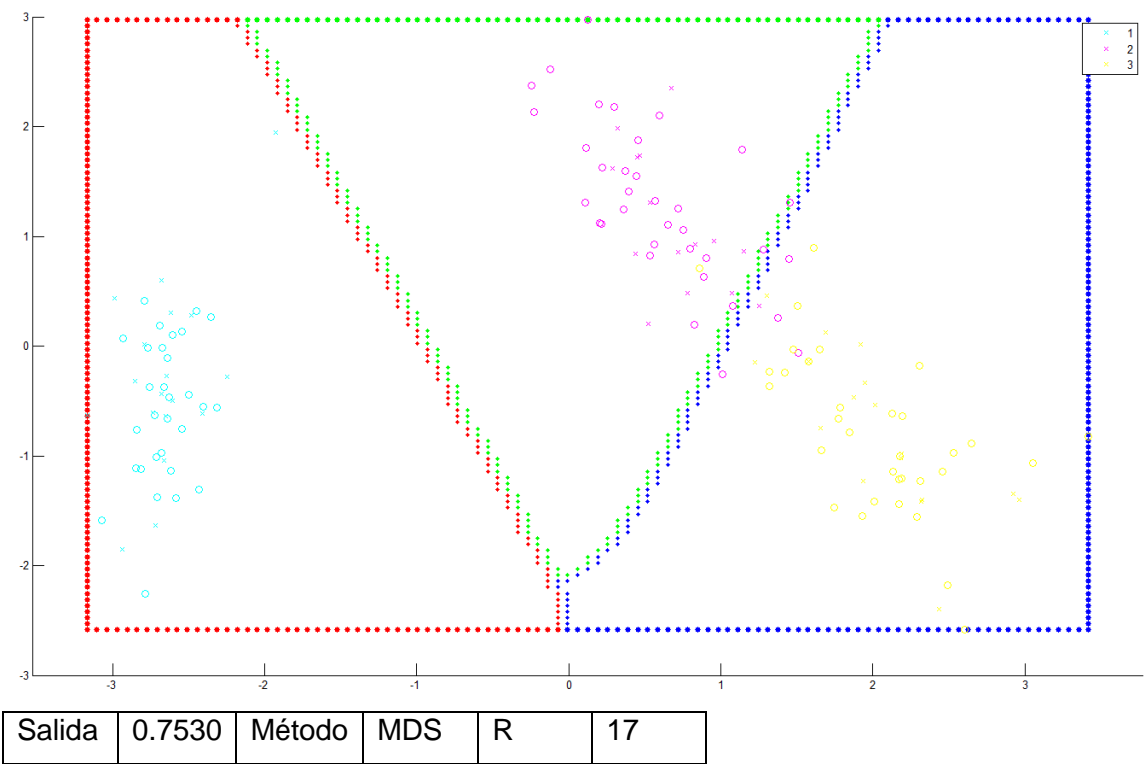
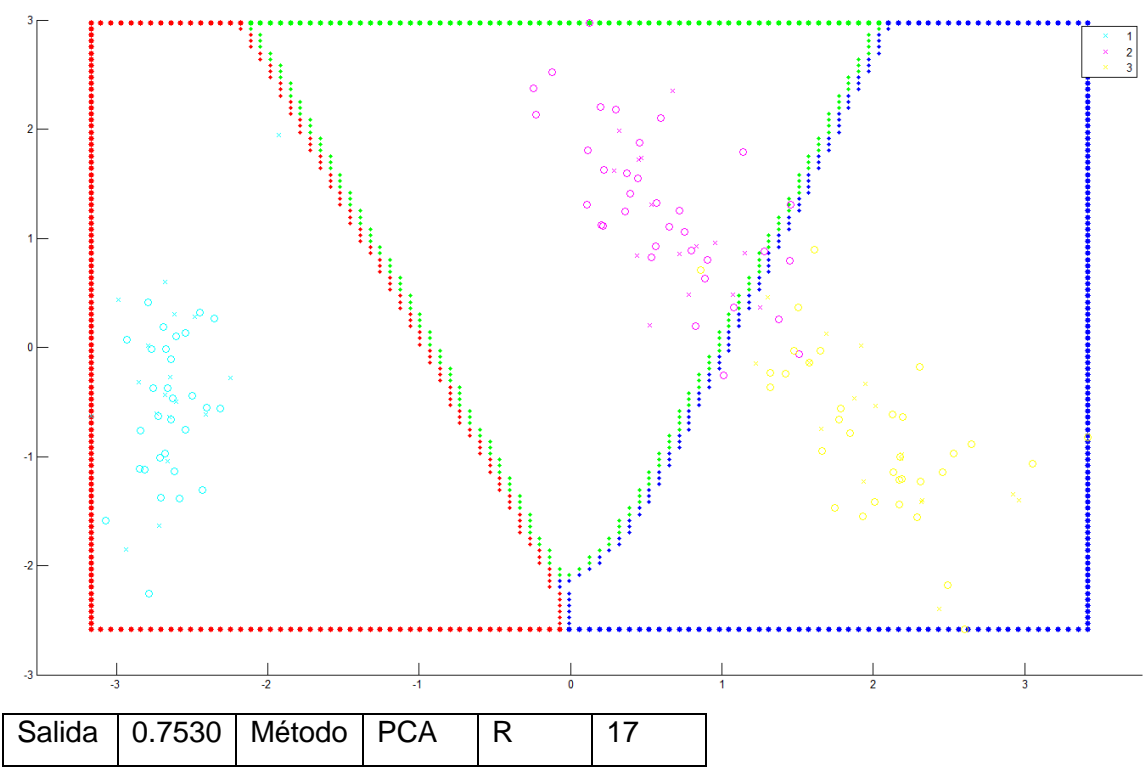
Salida	0.8146	Método	tSNE	R	10
--------	--------	--------	------	---	----

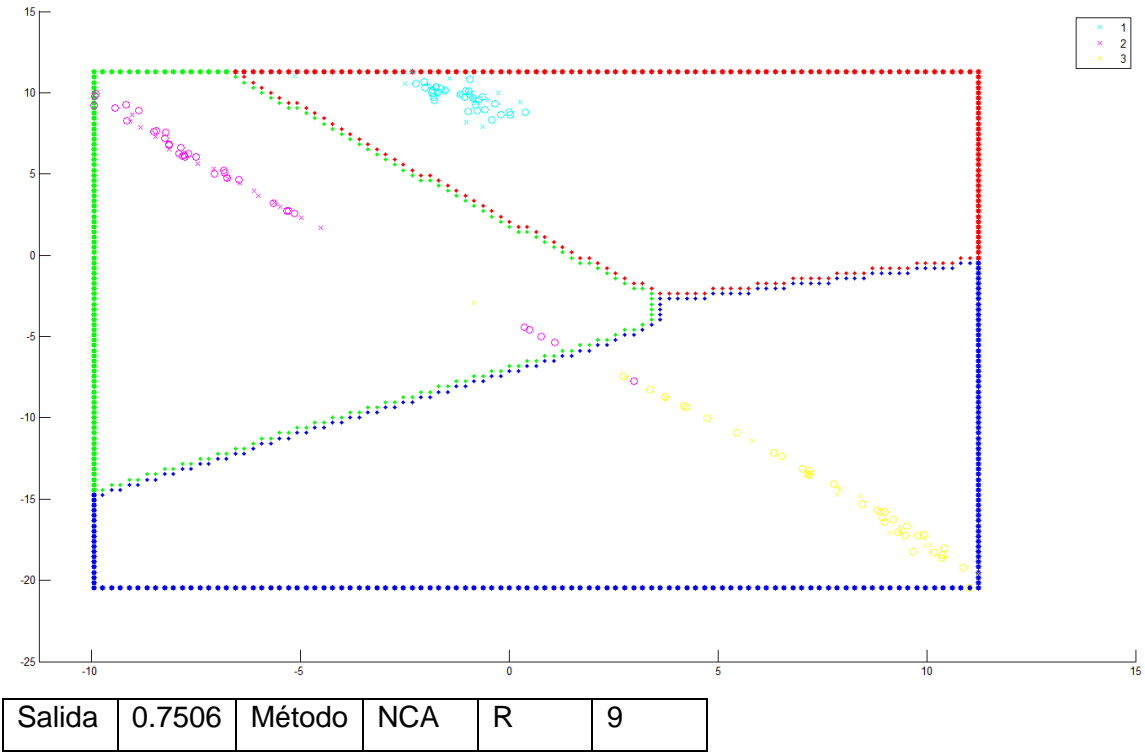
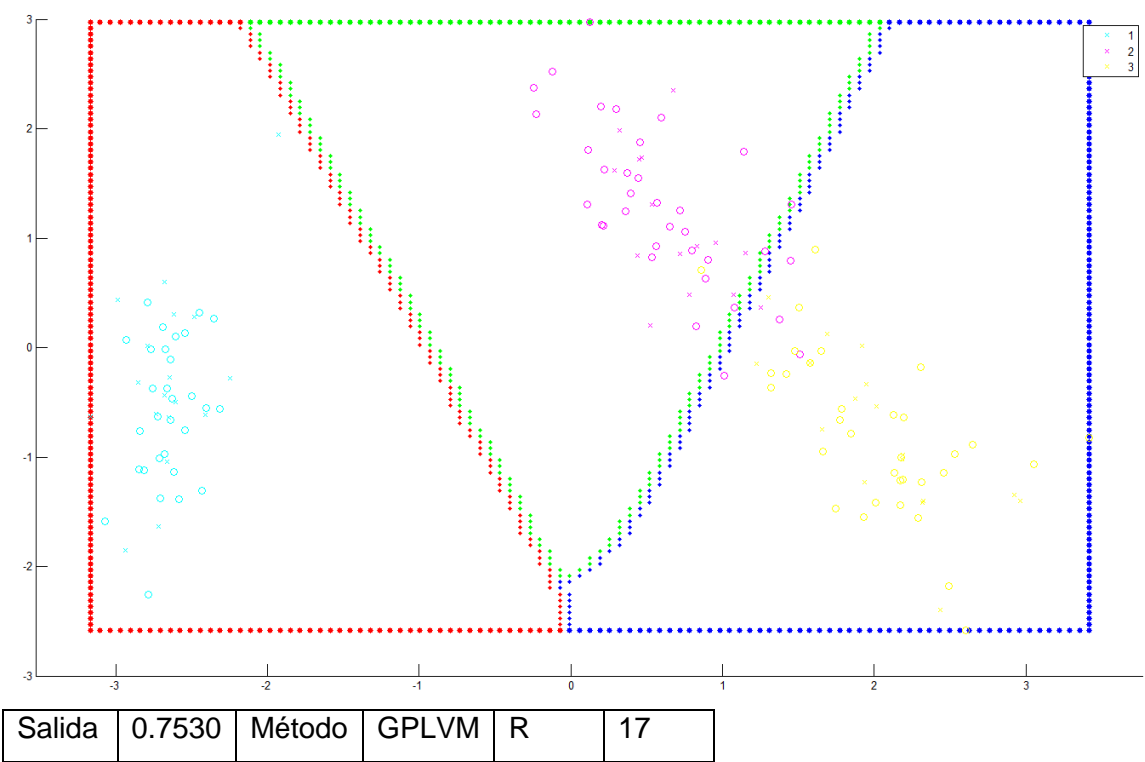
Visualización bidimensional de problemas de clasificación en alta dimensión



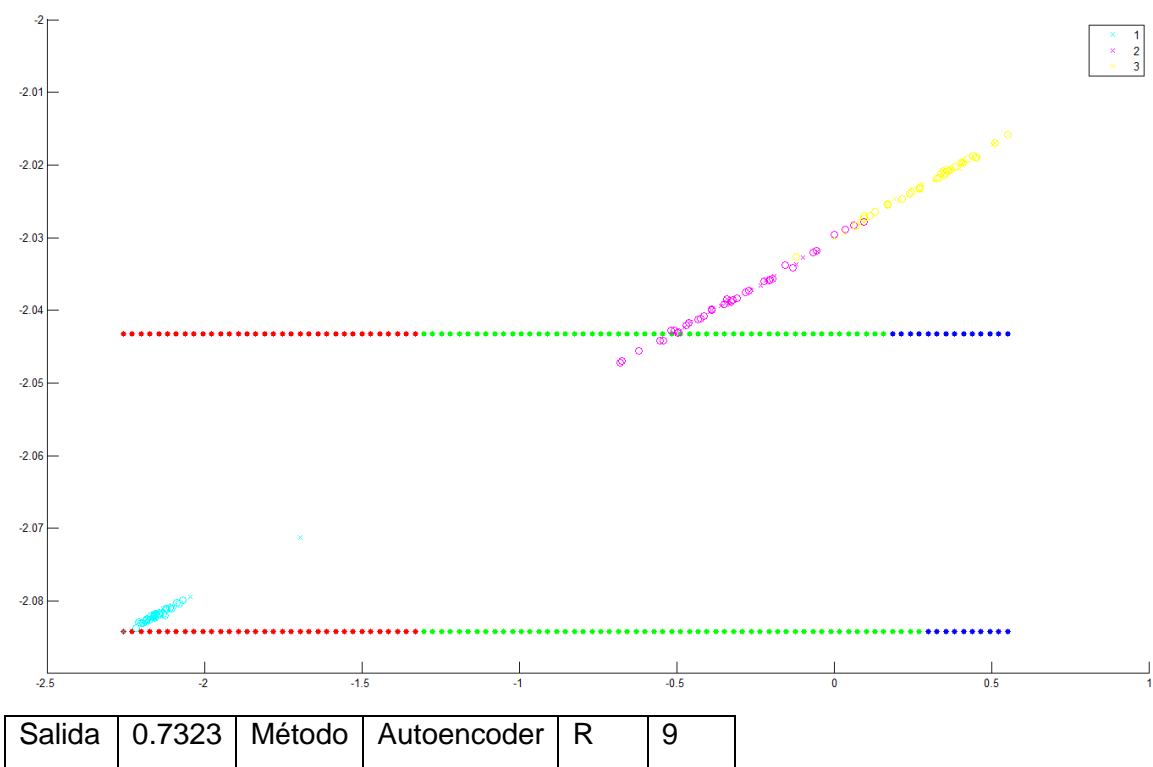
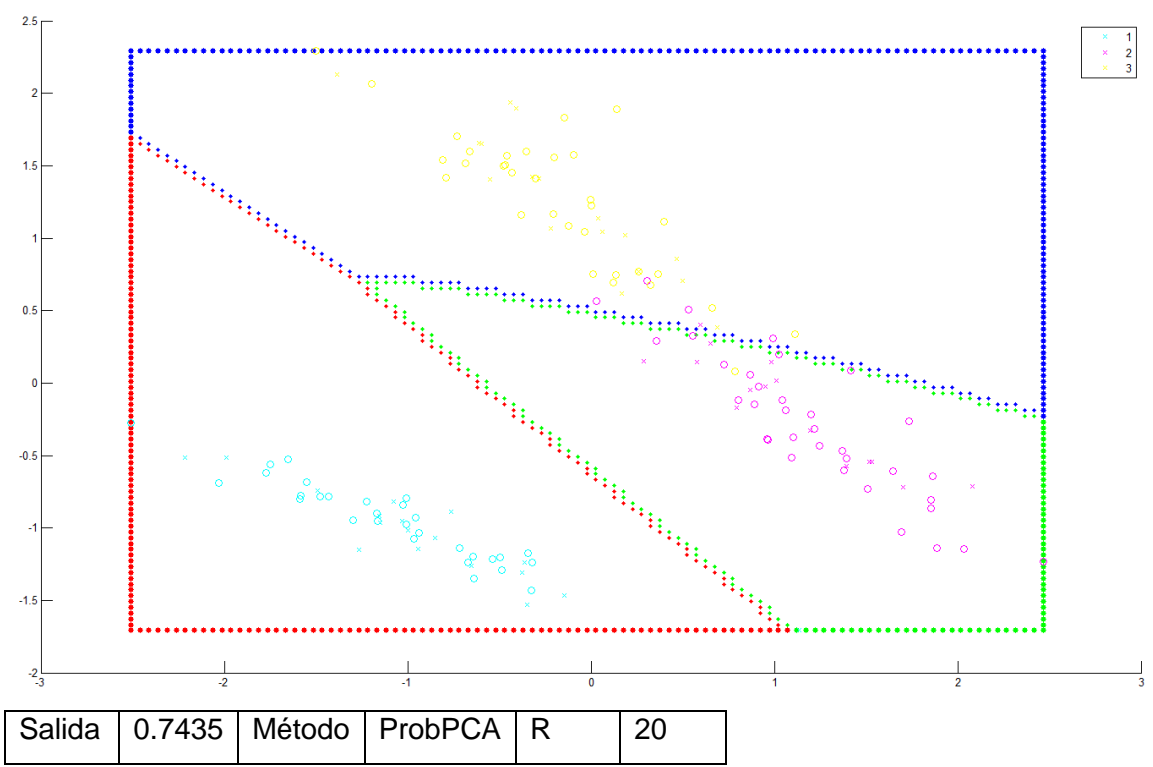


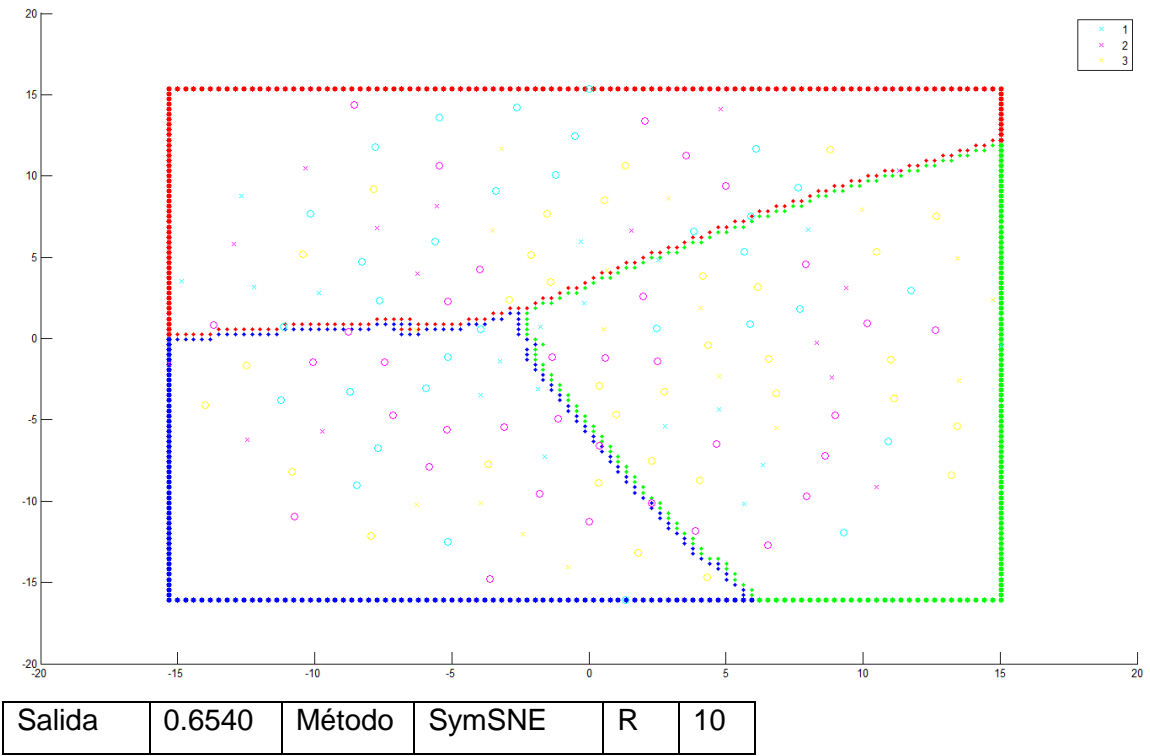
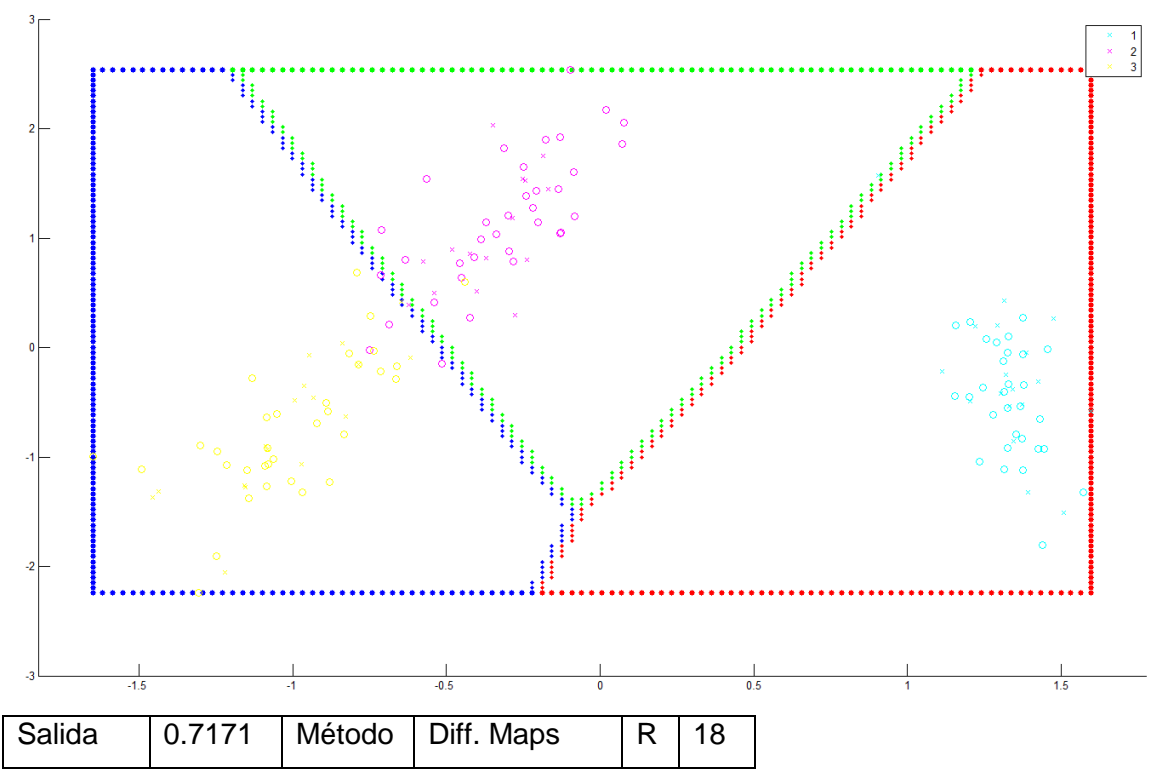
Visualización bidimensional de problemas de clasificación en alta dimensión



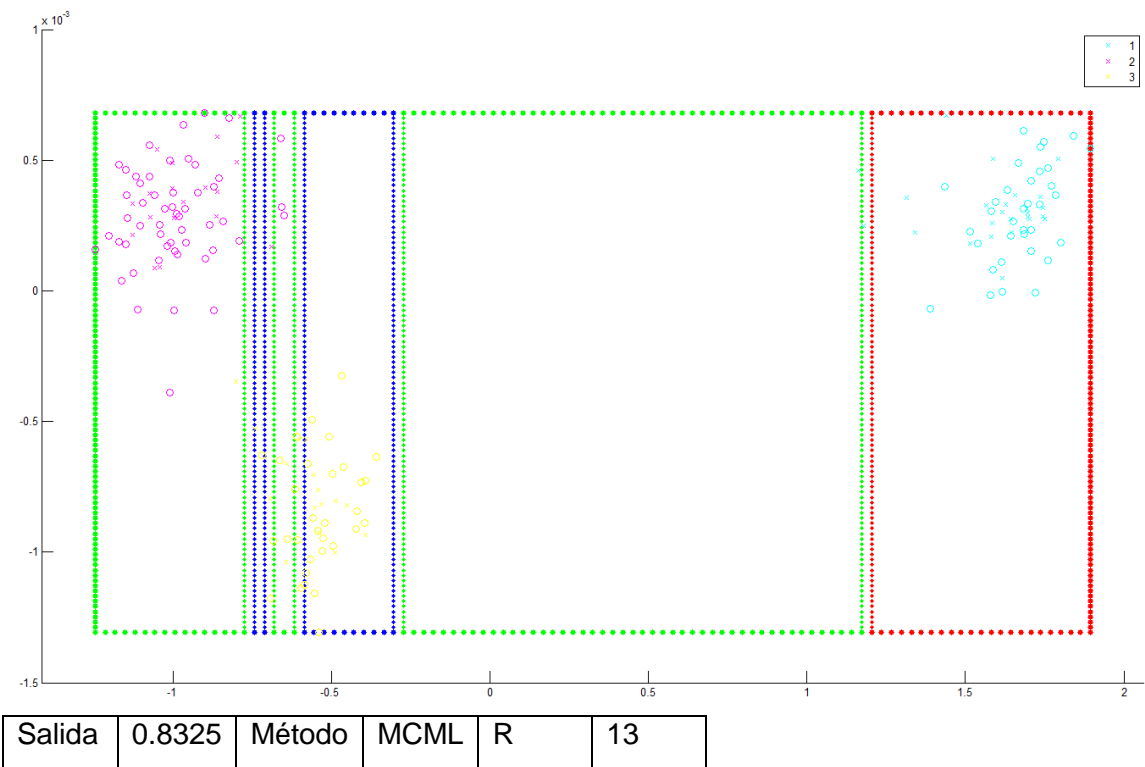
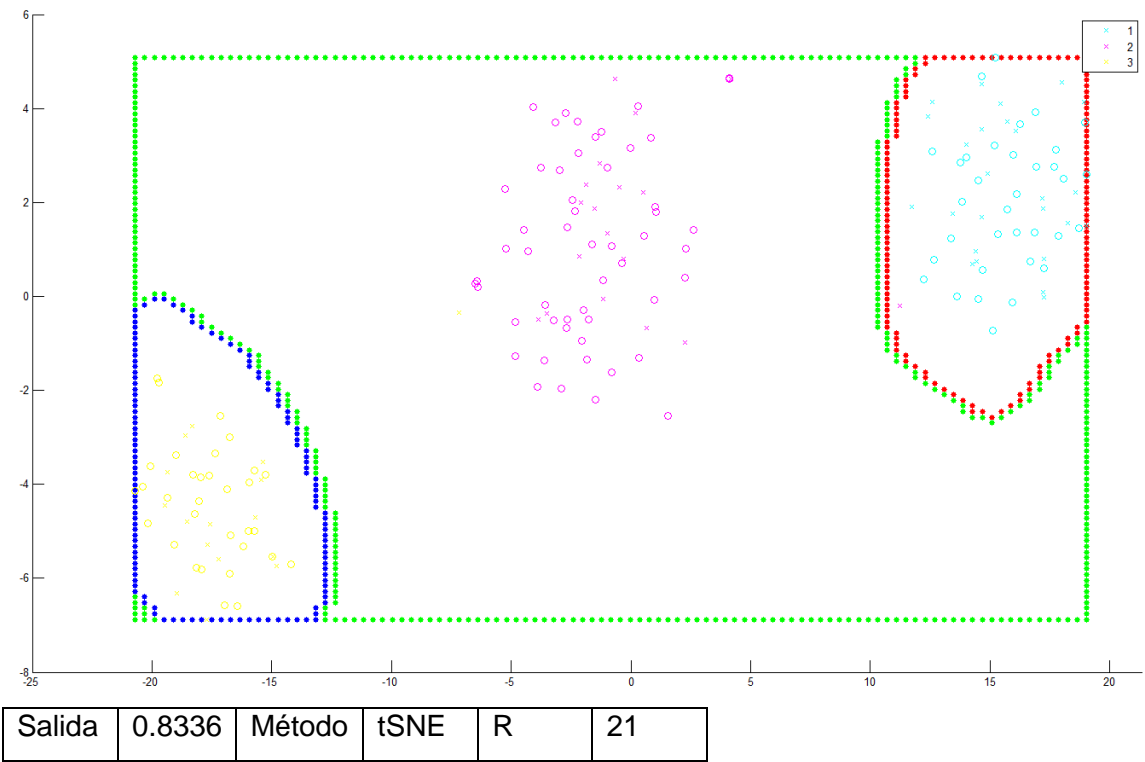


Visualización bidimensional de problemas de clasificación en alta dimensión

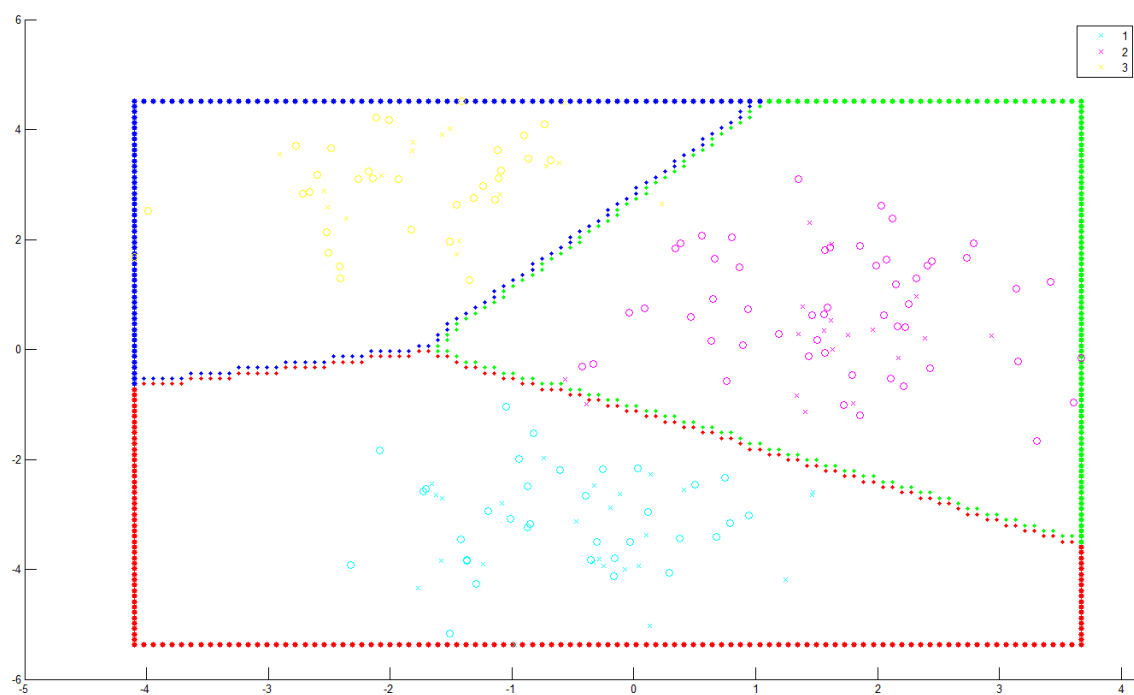




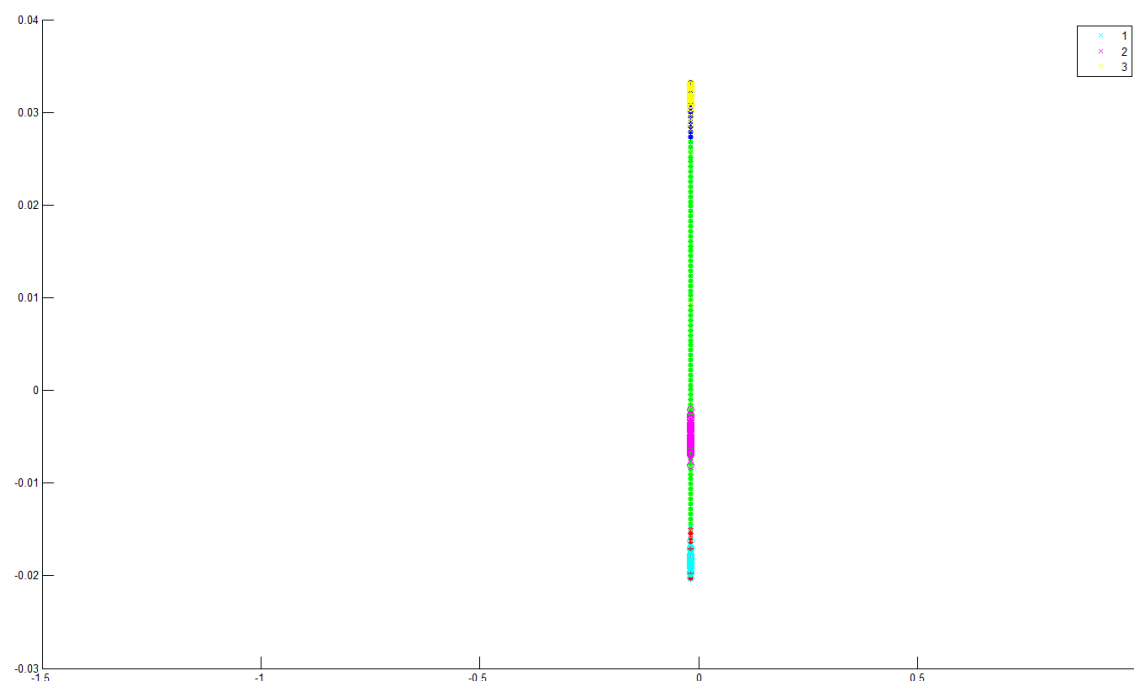
5.1.2 Conjunto Wine





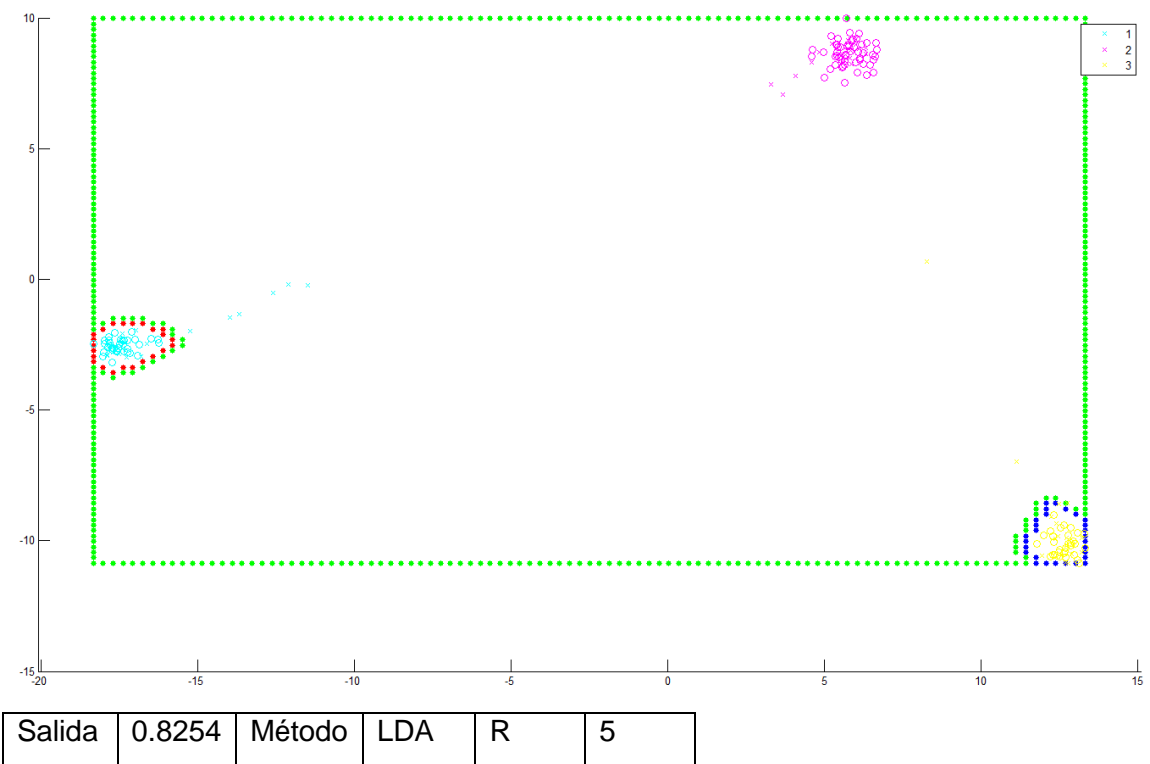
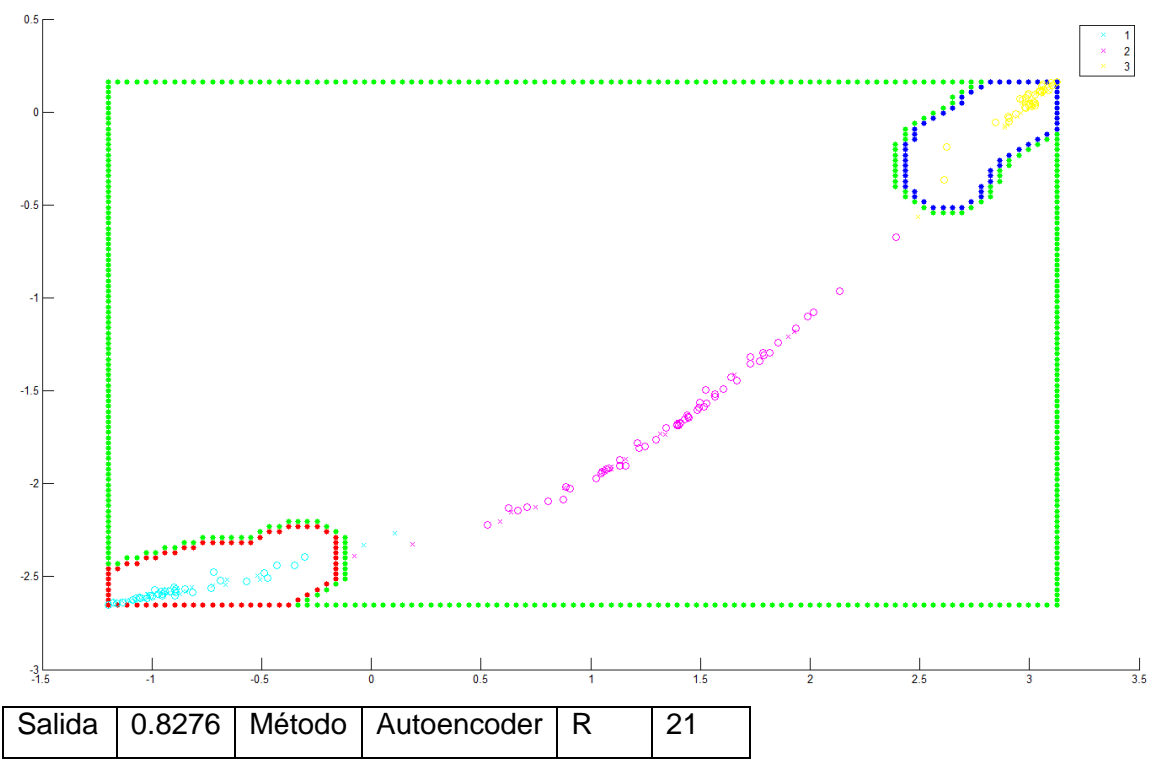


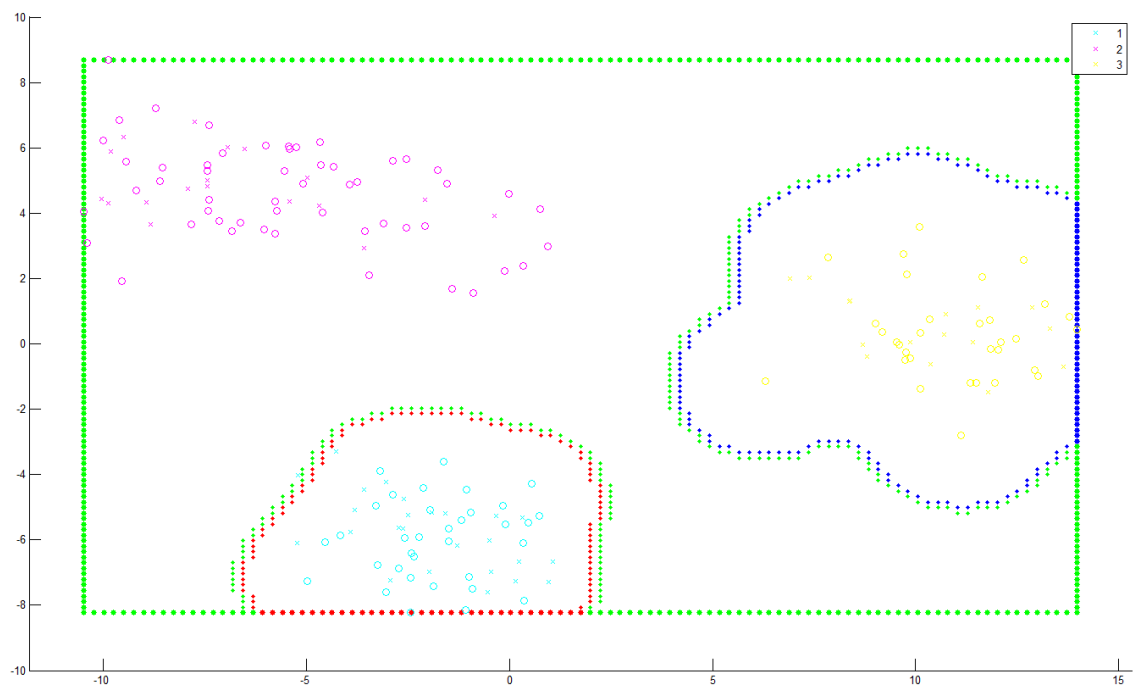
Salida	0.8298	Método	SNE	R	15
--------	--------	--------	-----	---	----



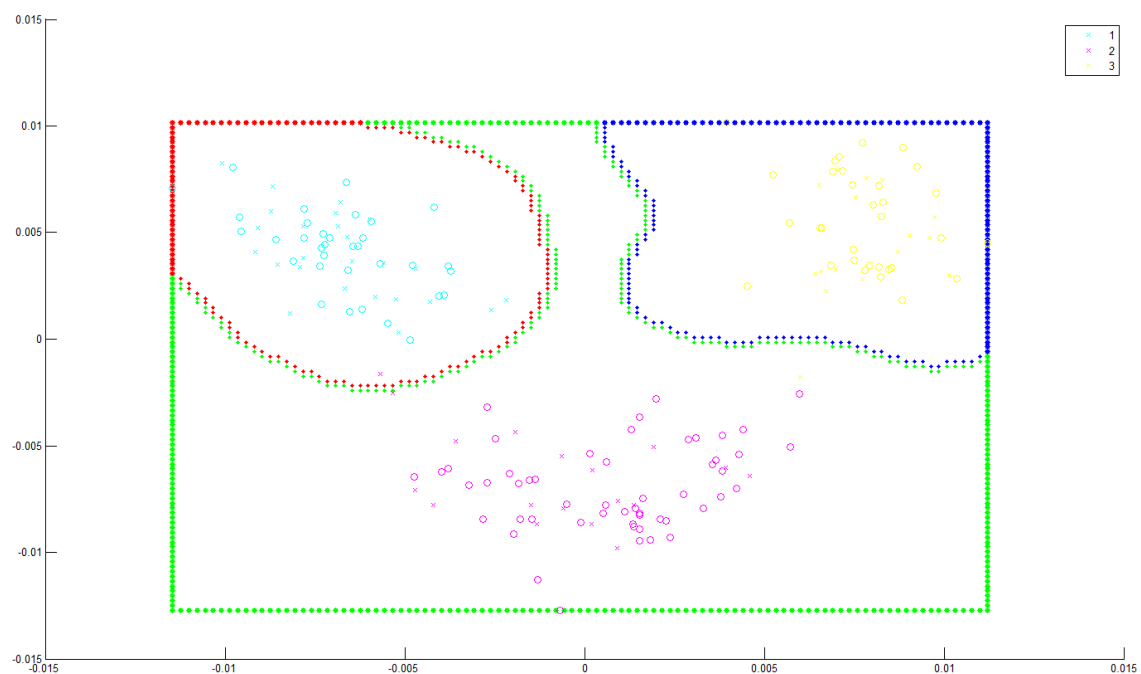
Salida	0.8283	Método	LPP	R	11
--------	--------	--------	-----	---	----

Visualización bidimensional de problemas de clasificación en alta dimensión



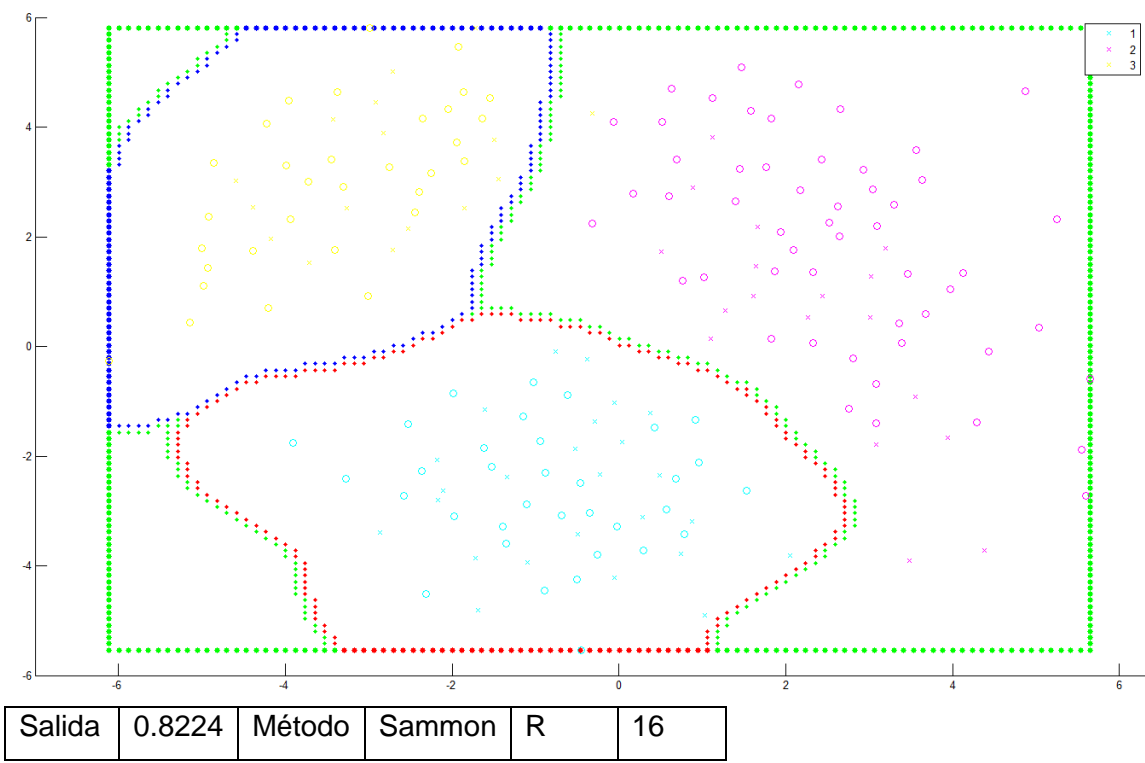
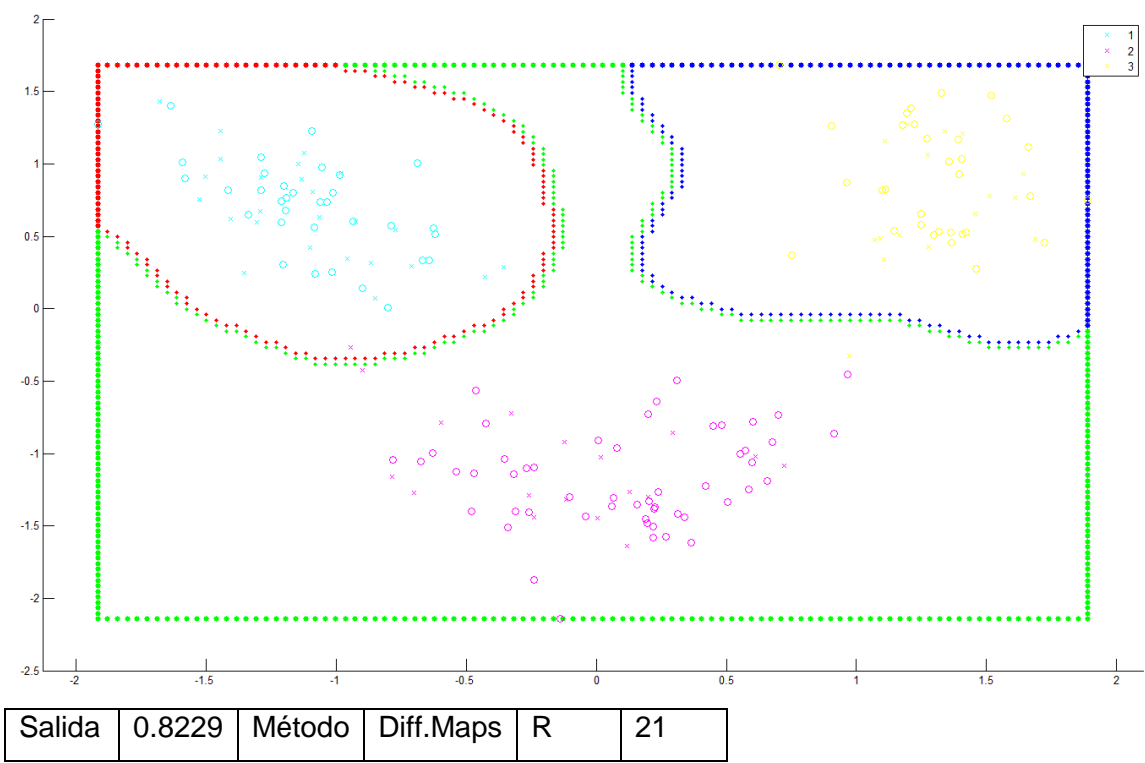


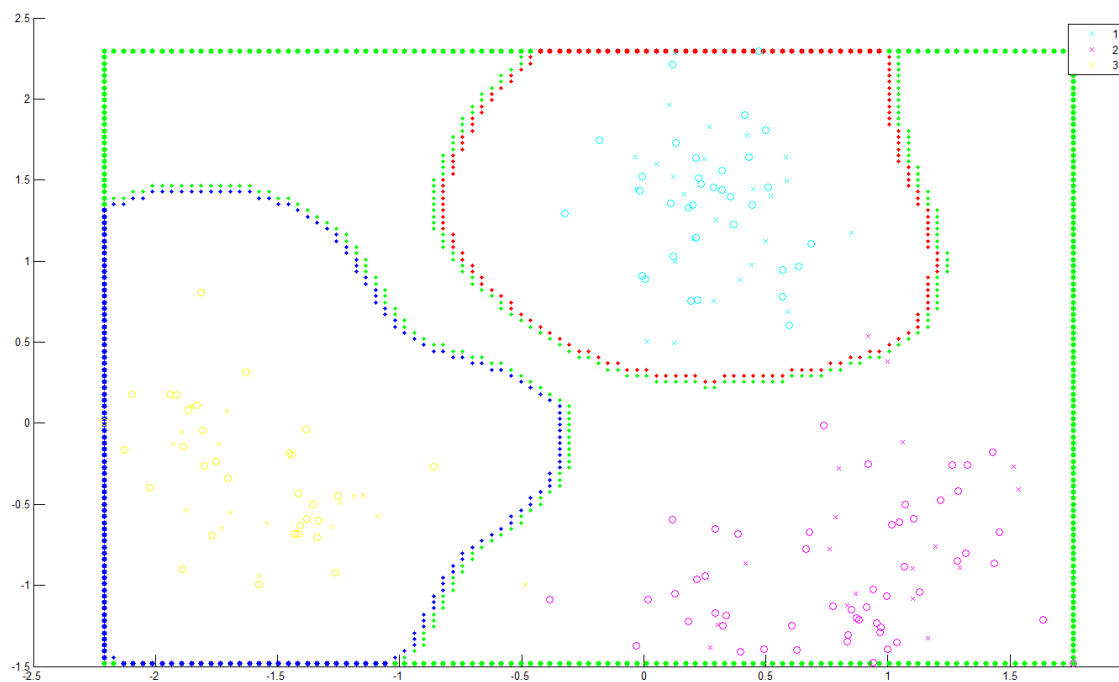
Salida	0.8235	Método	NCA	R	17
--------	--------	--------	-----	---	----



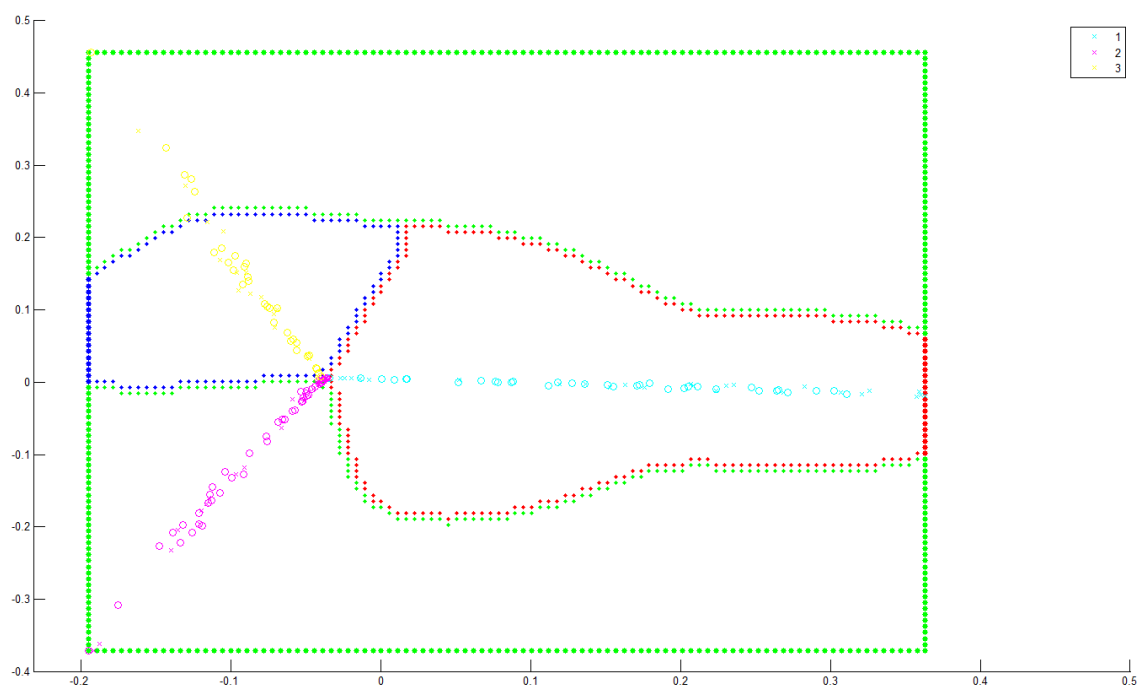
Salida	0.8232	Método	Laplacian	R	21
--------	--------	--------	-----------	---	----

Visualización bidimensional de problemas de clasificación en alta dimensión



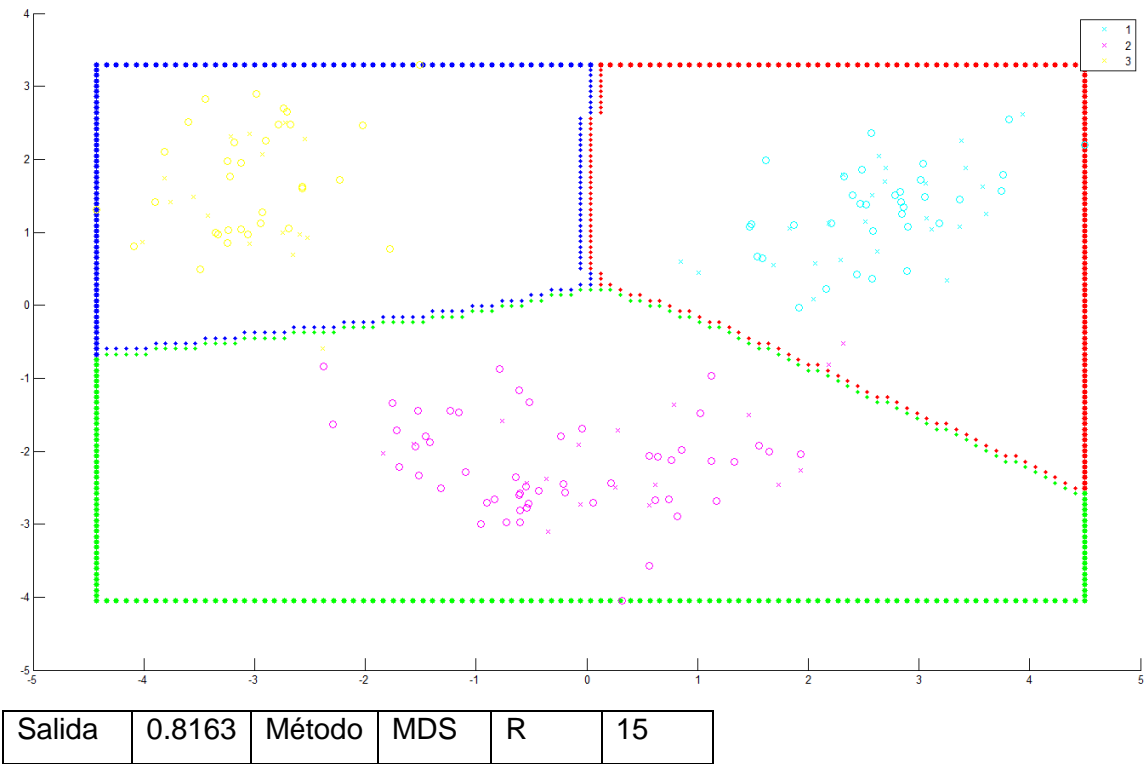
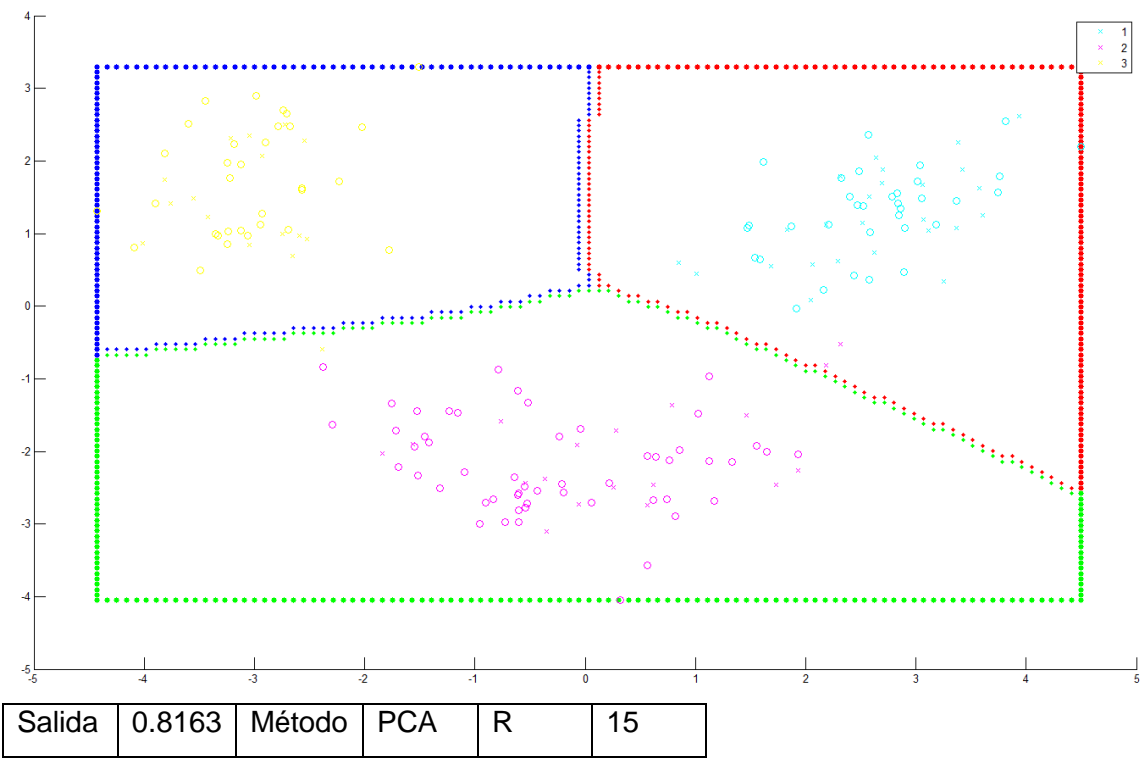


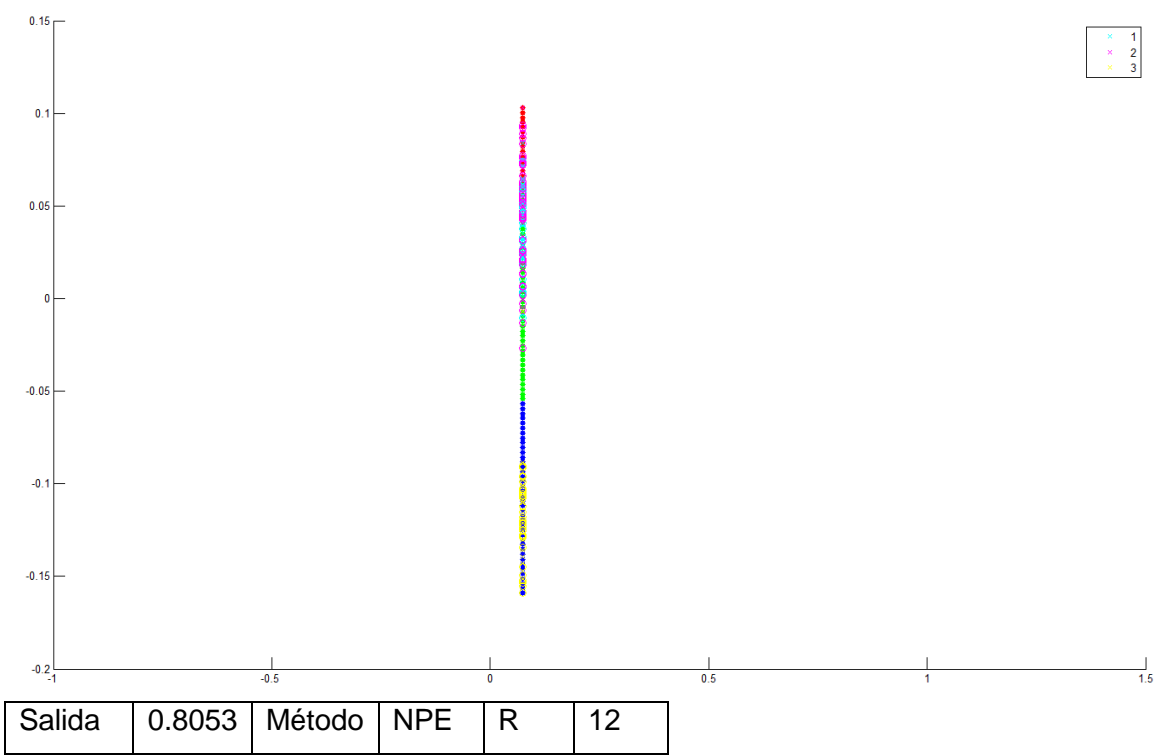
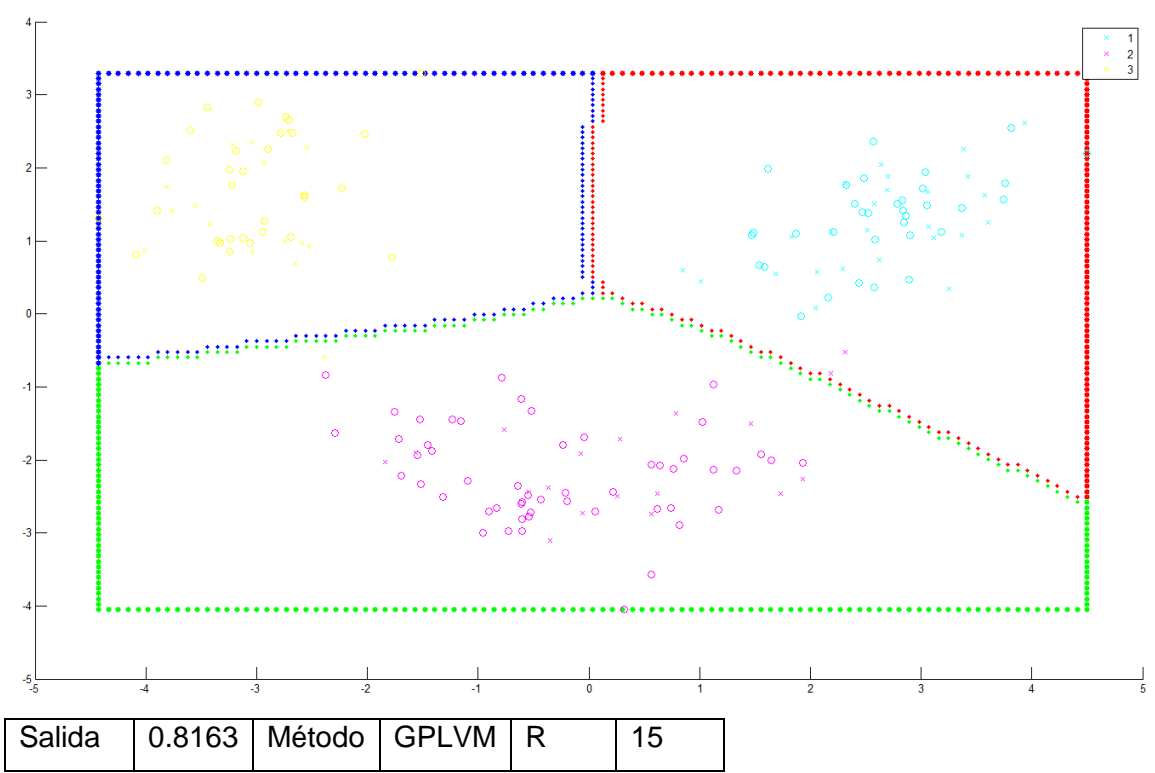
Salida	0.8224	Método	ProbPCA	R	15
--------	--------	--------	---------	---	----



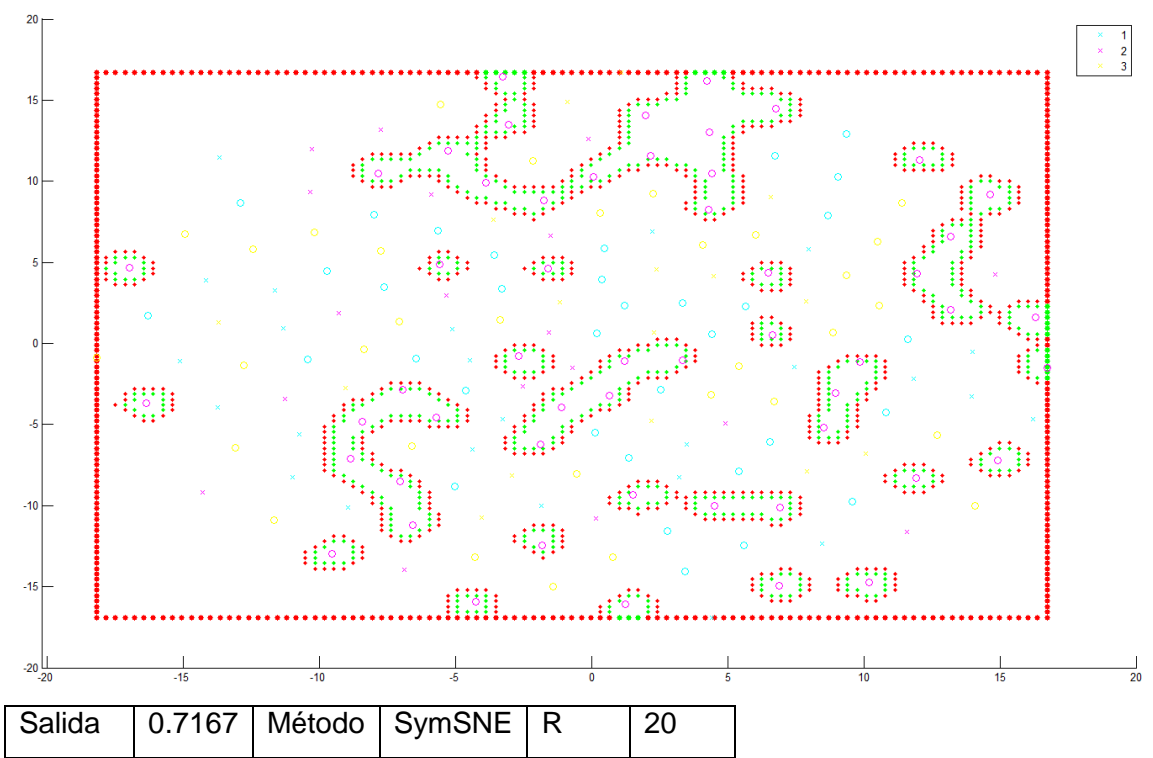
Salida	0.8180	Método	KernelPCA	R	19
--------	--------	--------	-----------	---	----

Visualización bidimensional de problemas de clasificación en alta dimensión

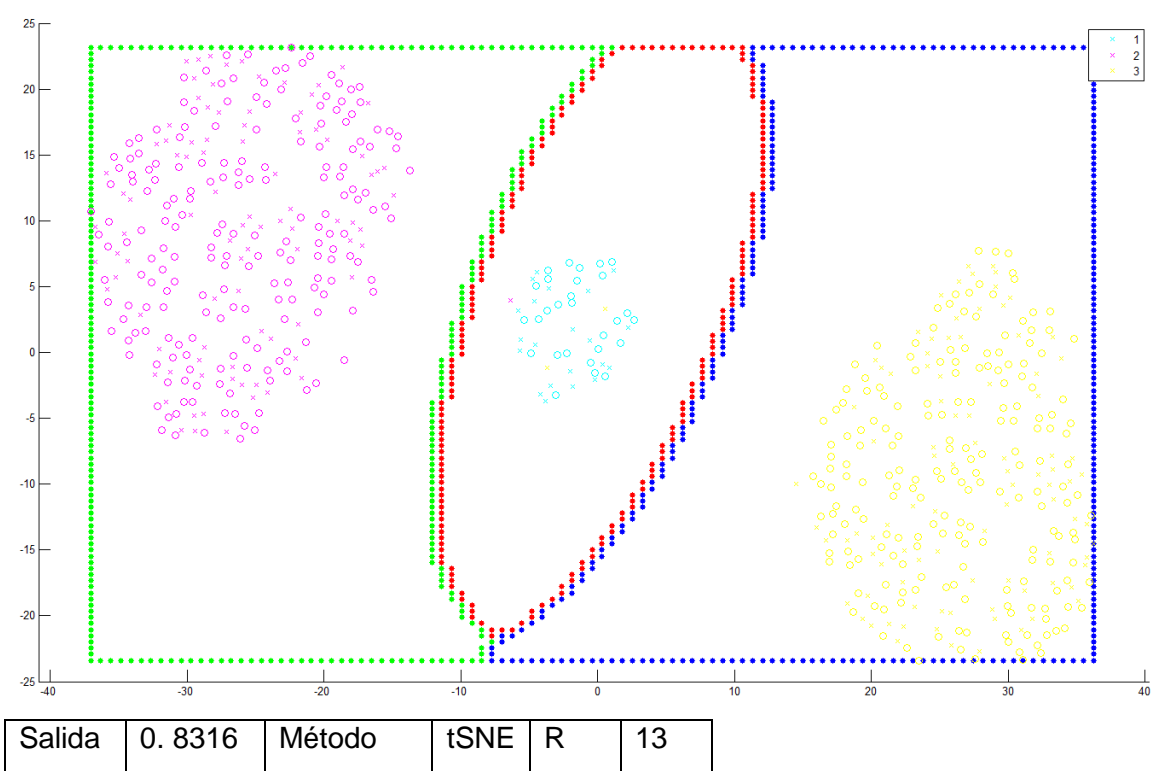




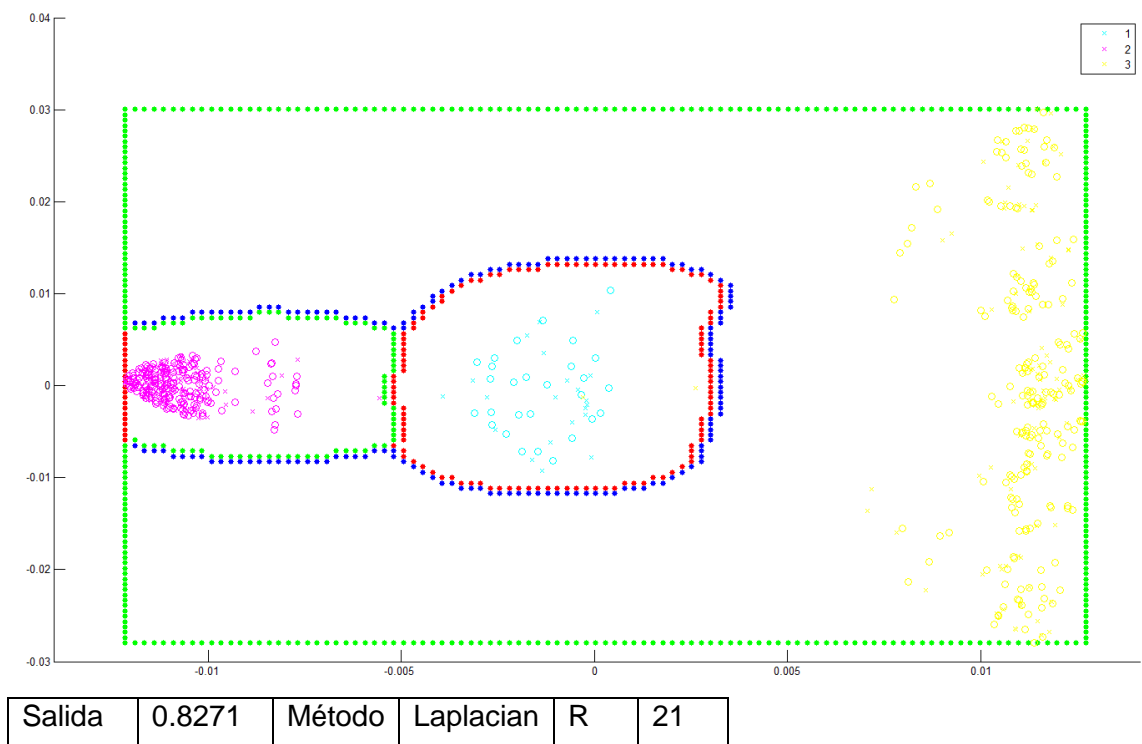
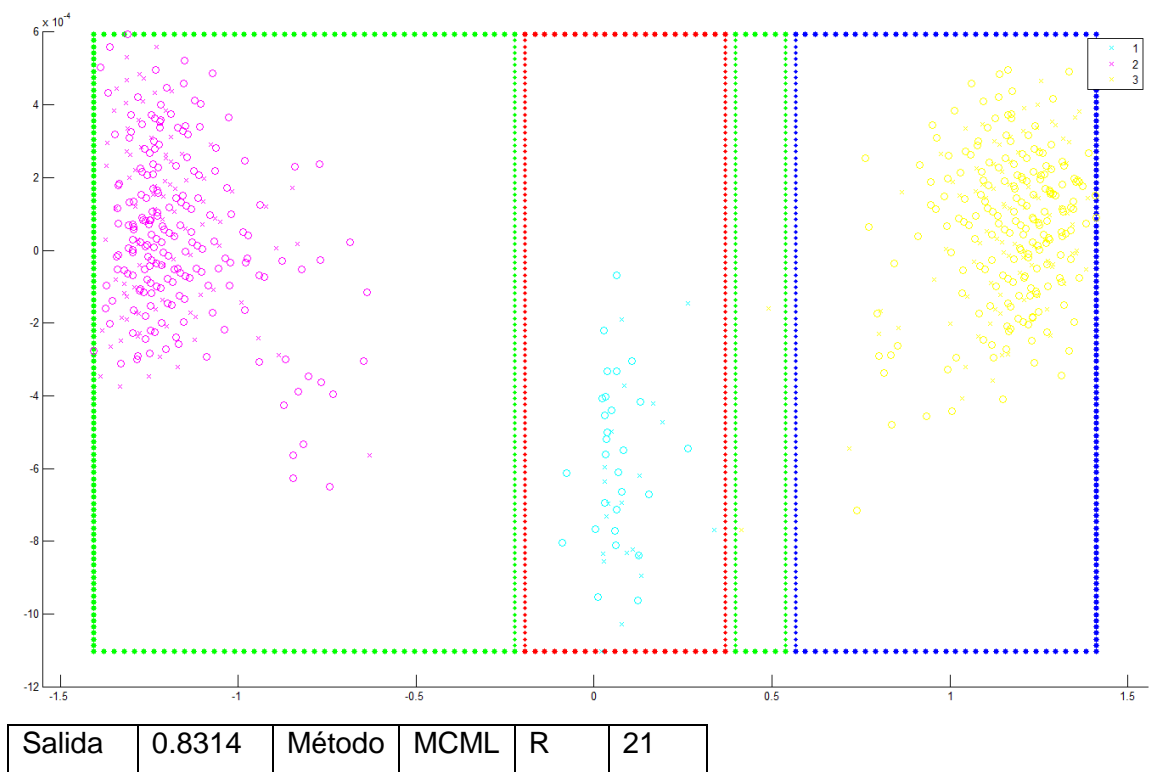
Visualización bidimensional de problemas de clasificación en alta dimensión



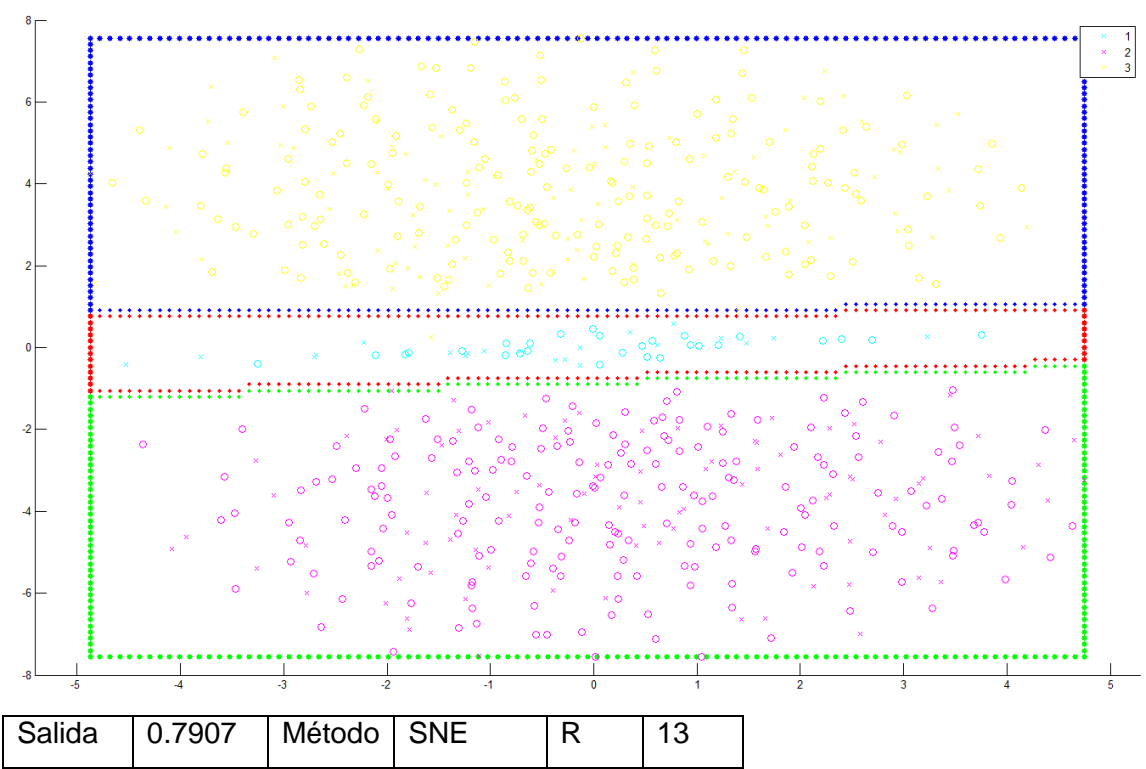
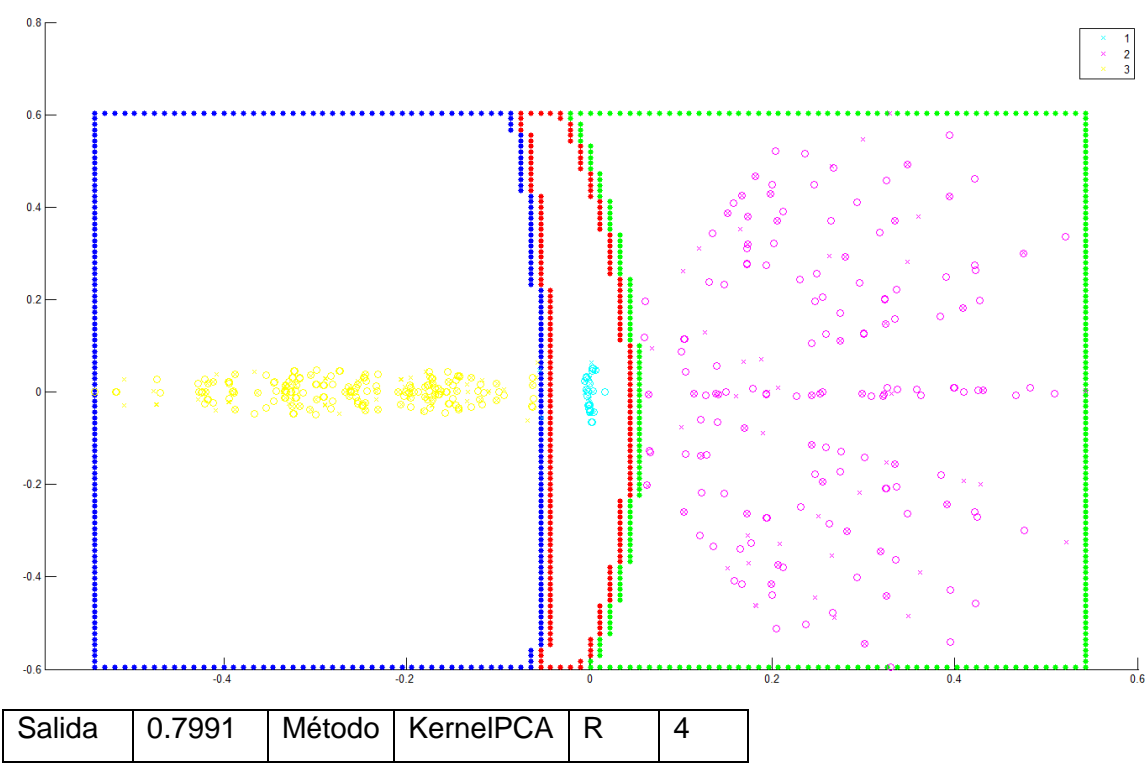
5.1.3 Conjunto Balance

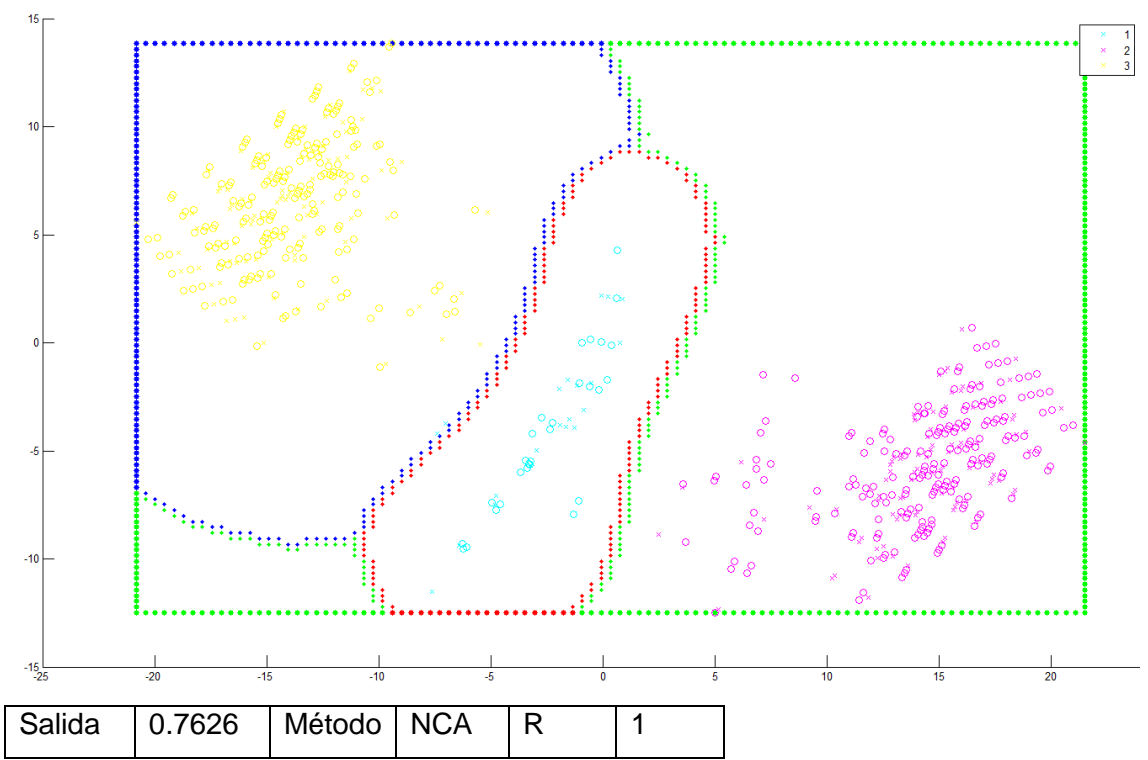
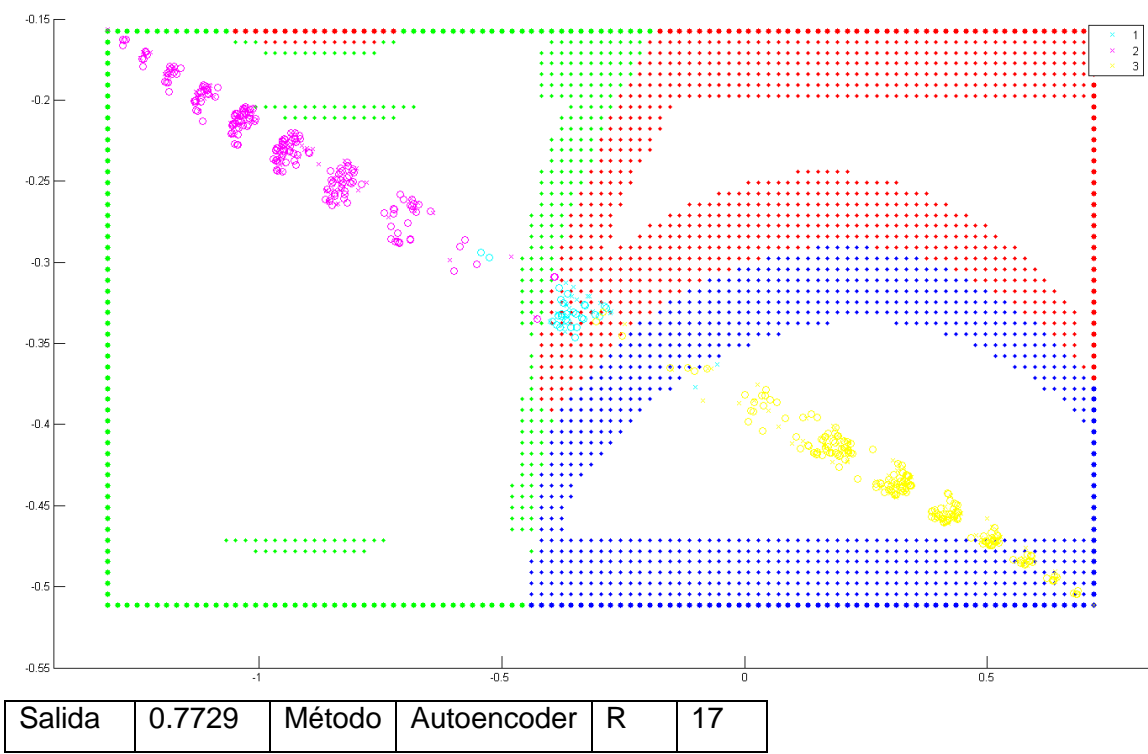




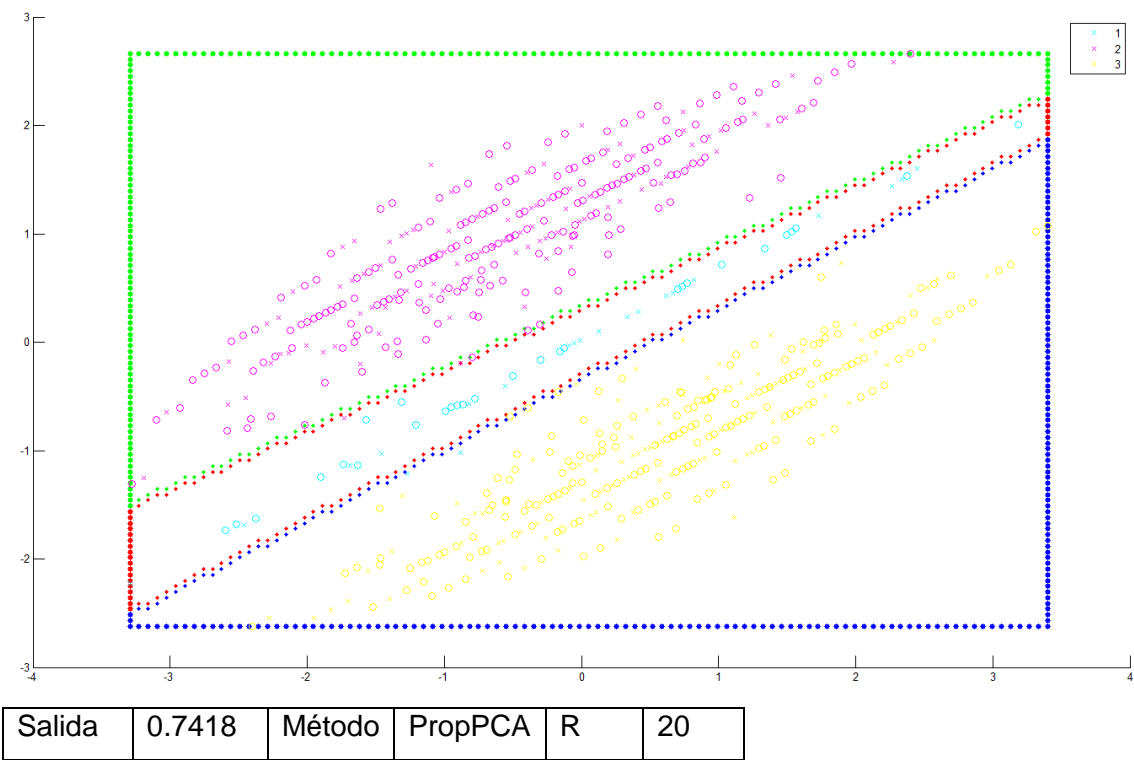
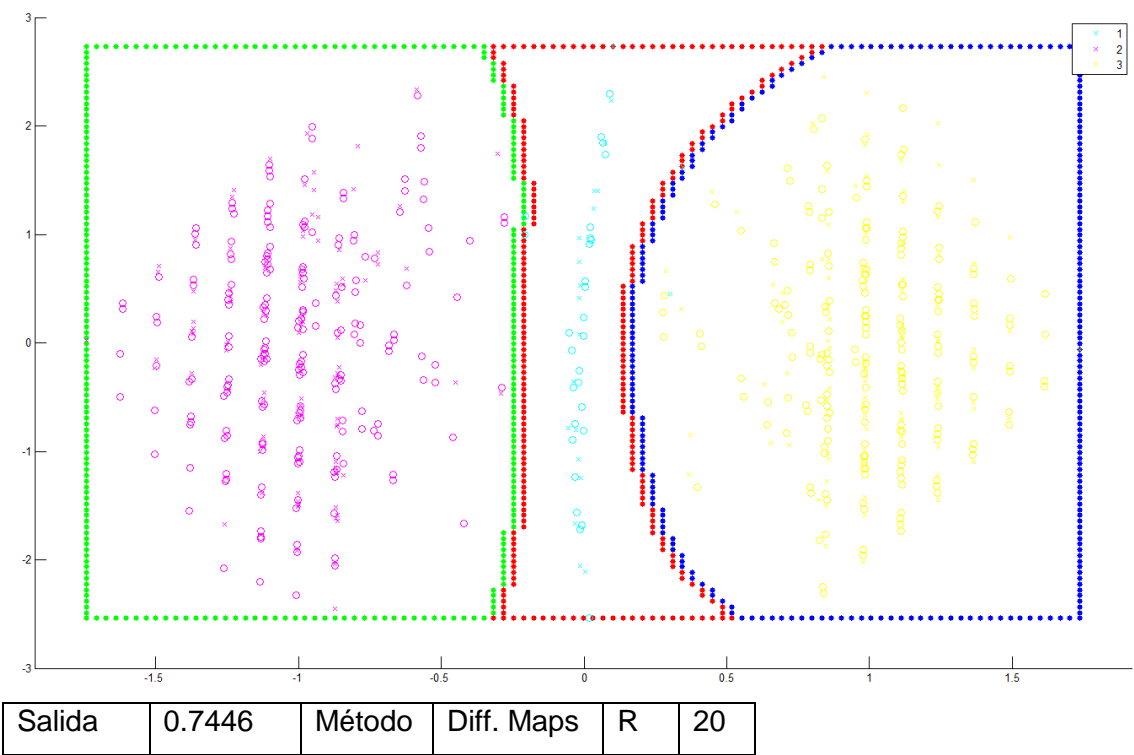


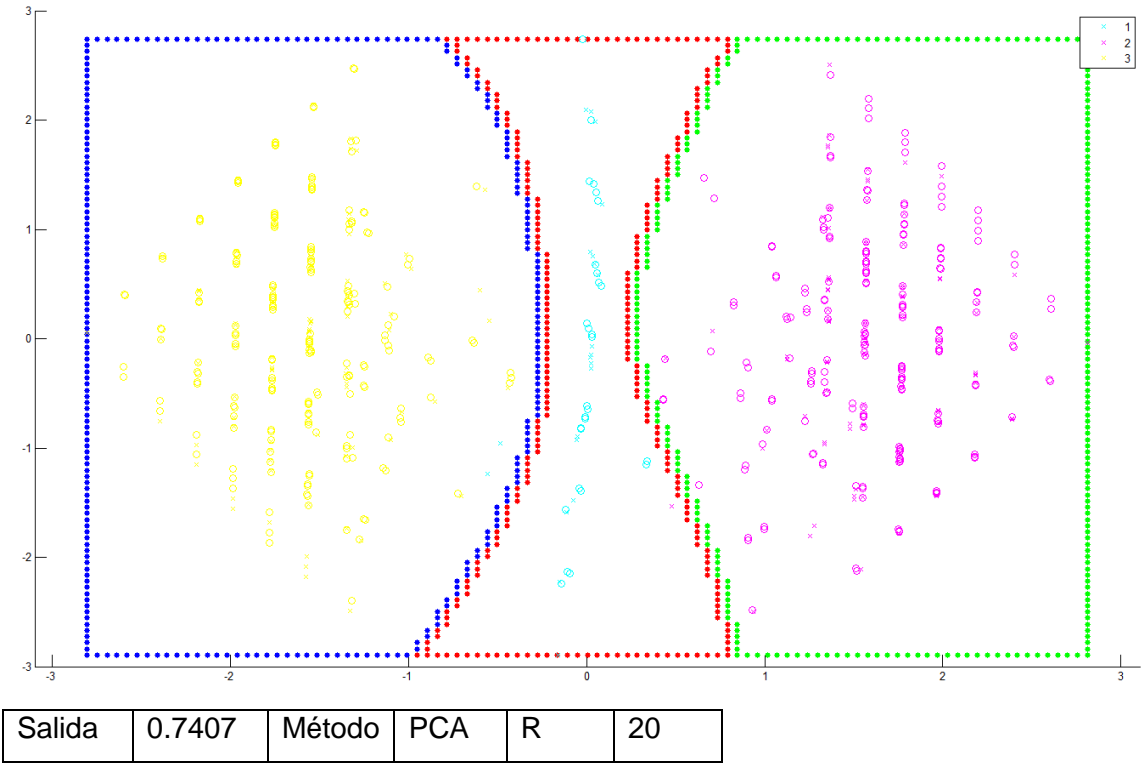
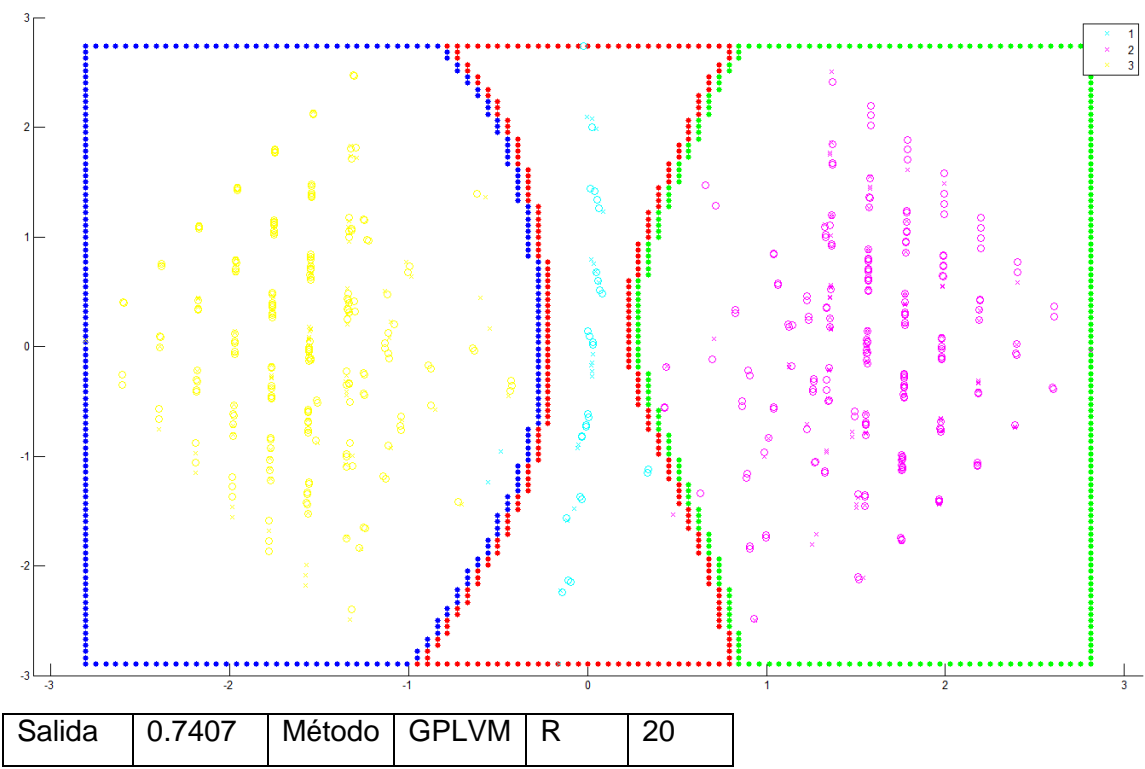
Visualización bidimensional de problemas de clasificación en alta dimensión



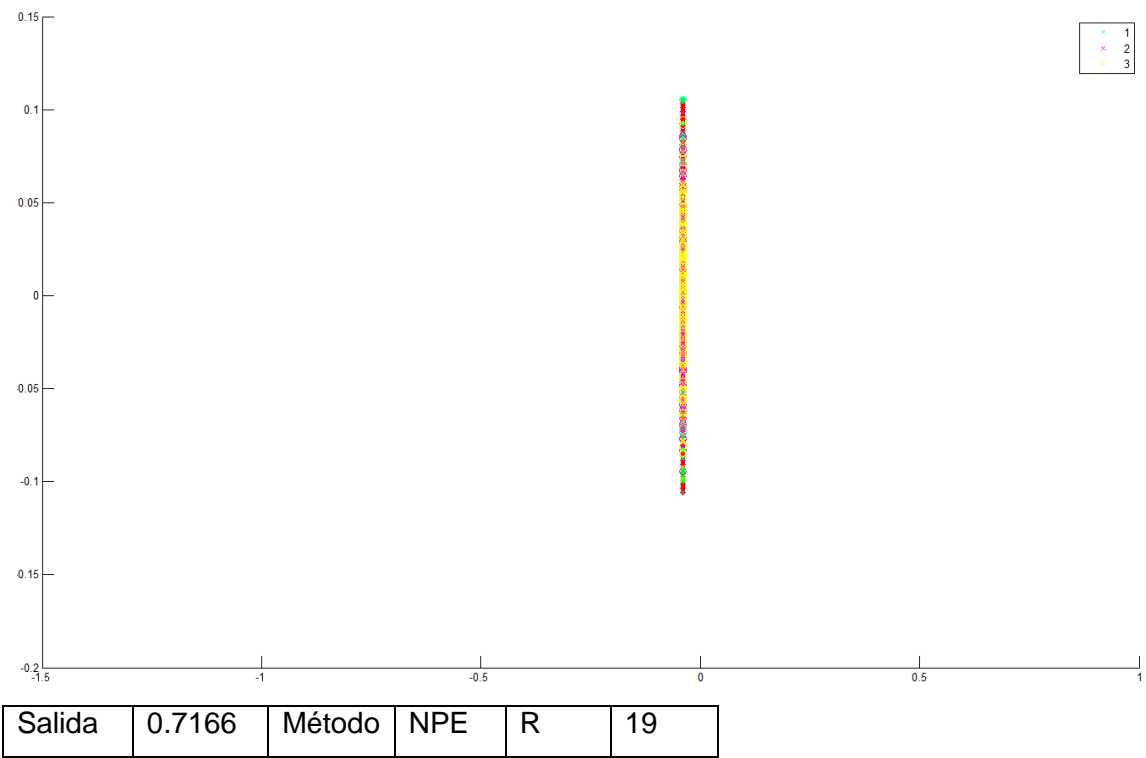
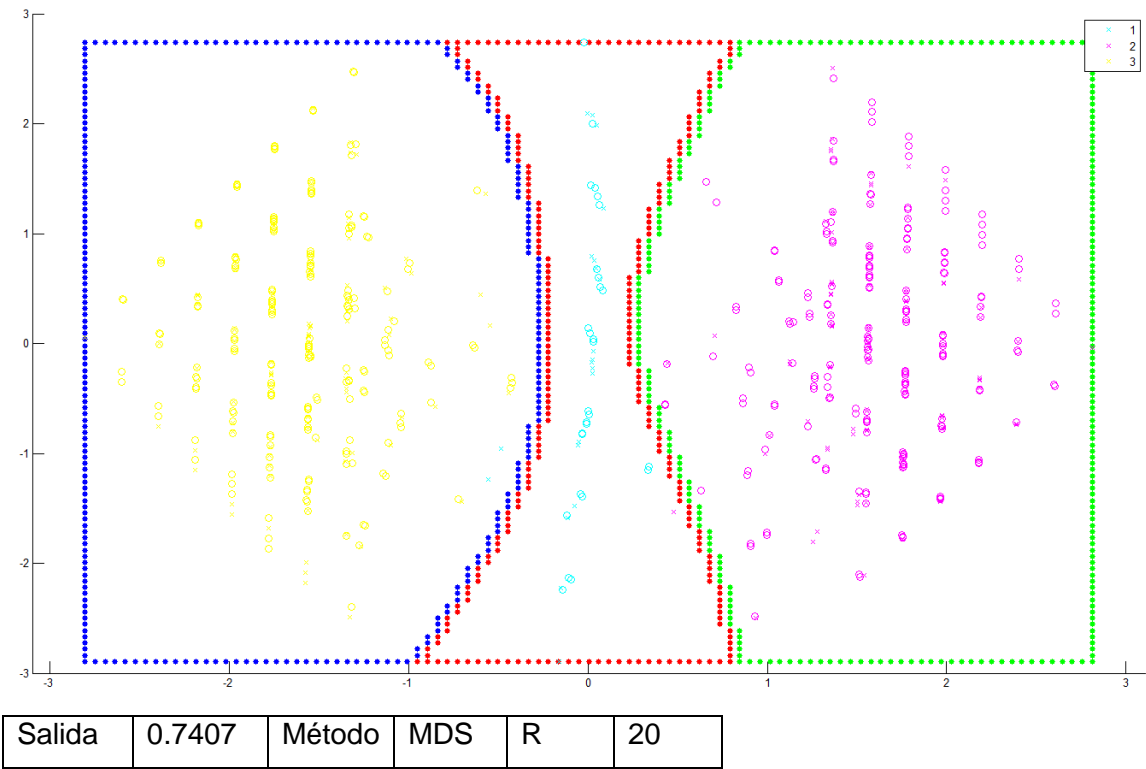


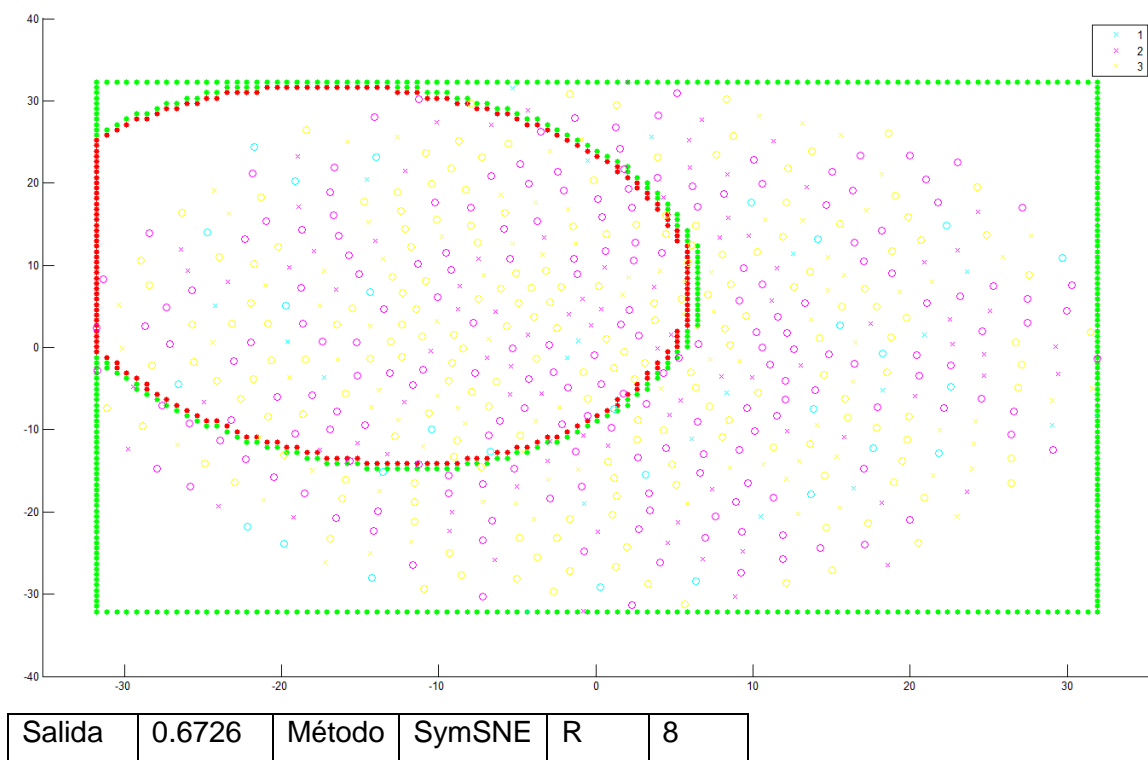
Visualización bidimensional de problemas de clasificación en alta dimensión



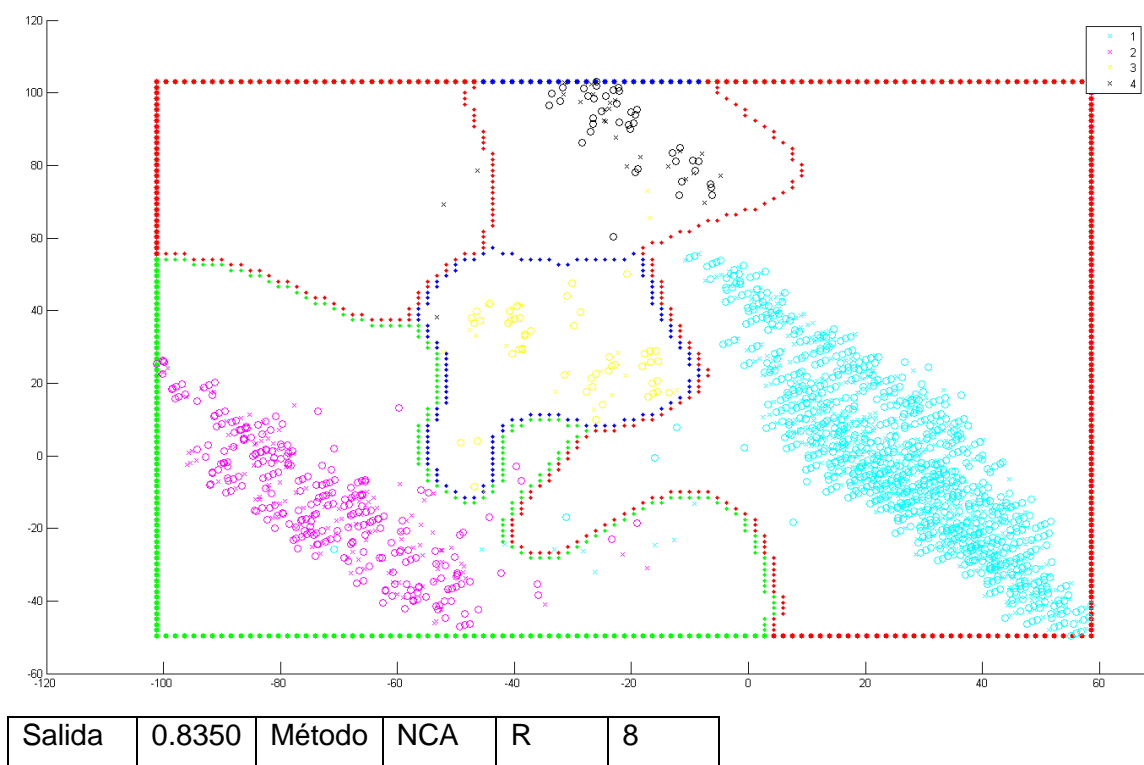


Visualización bidimensional de problemas de clasificación en alta dimensión

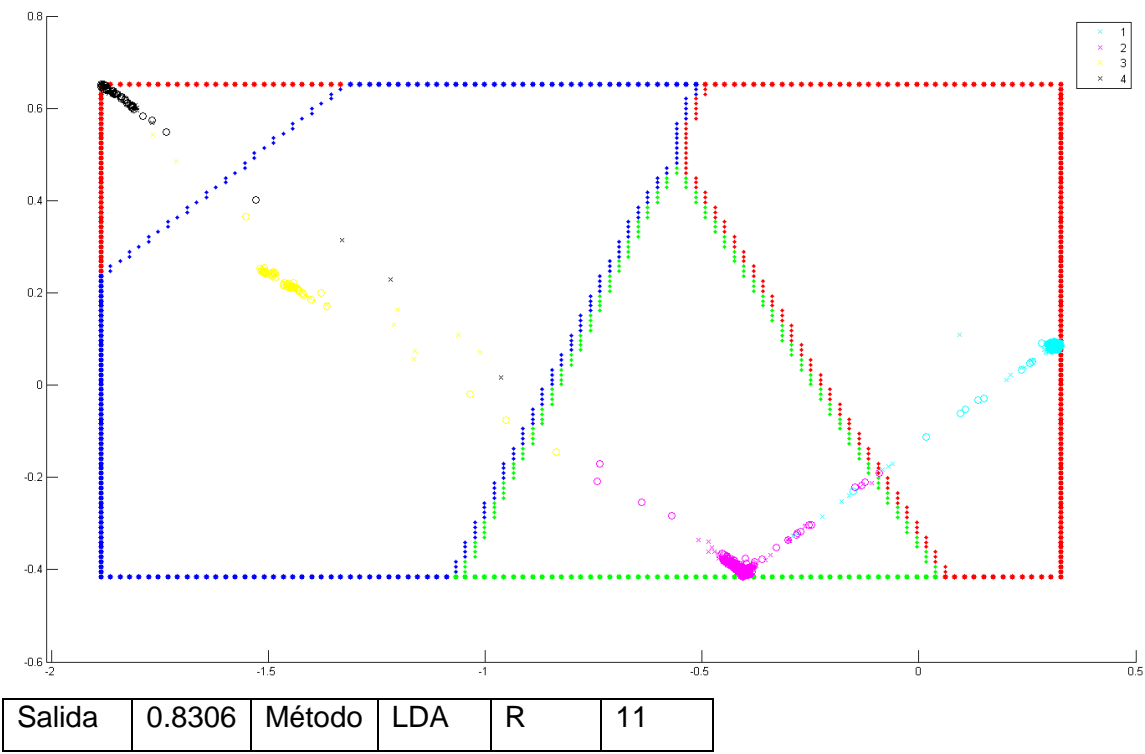
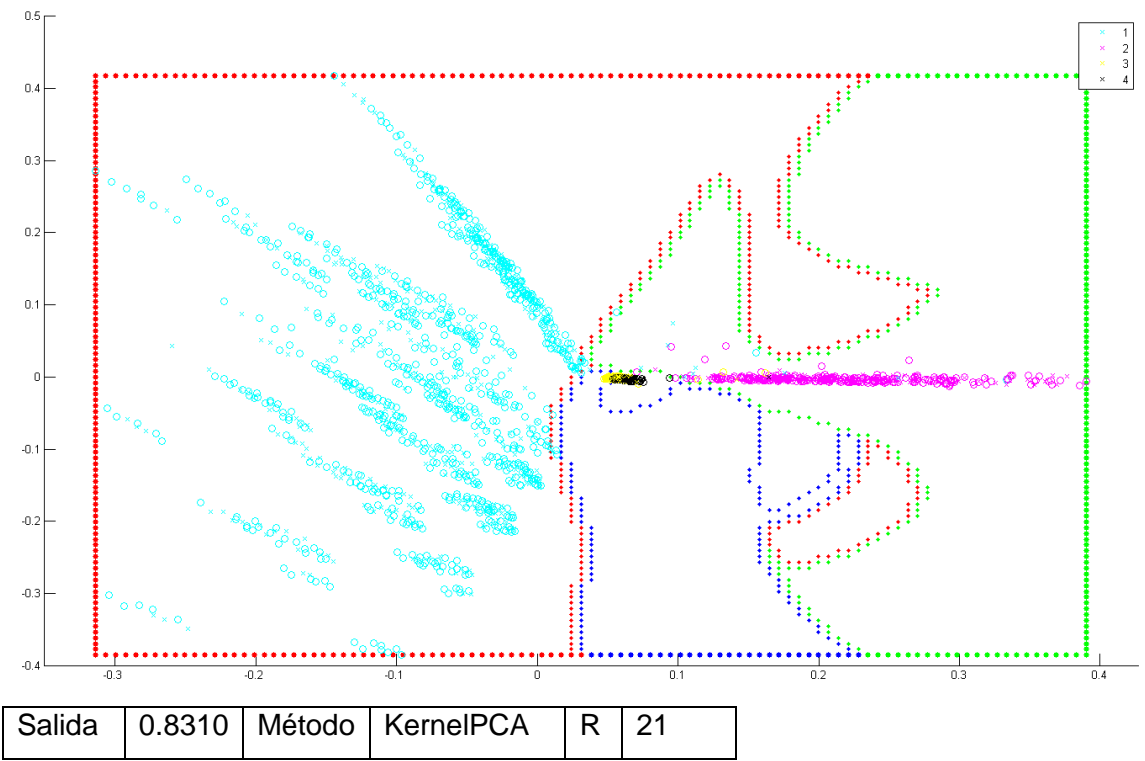




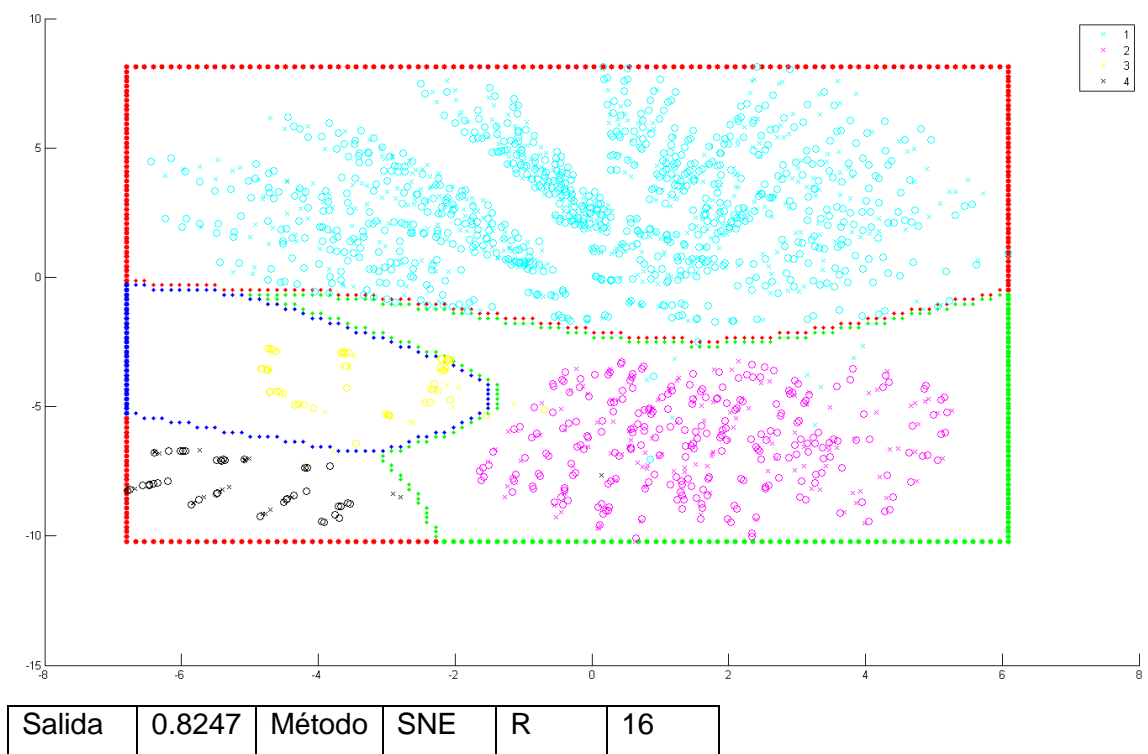
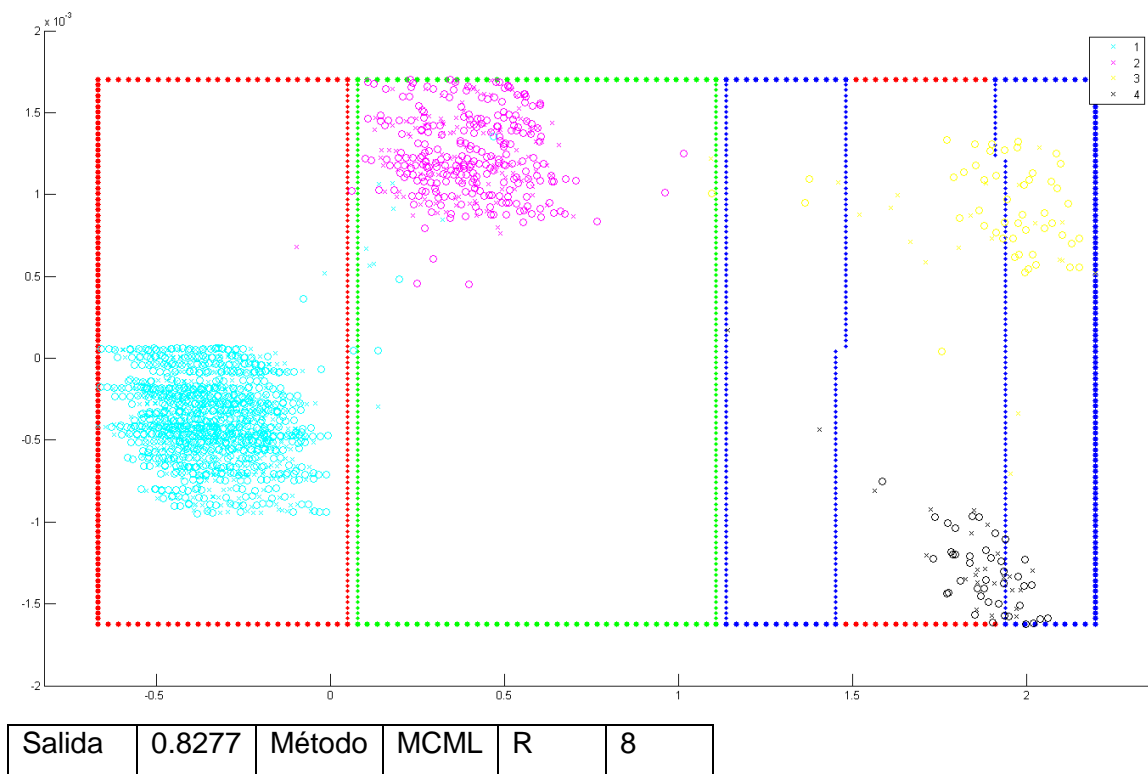
#### 5.1.4 Conjunto Car



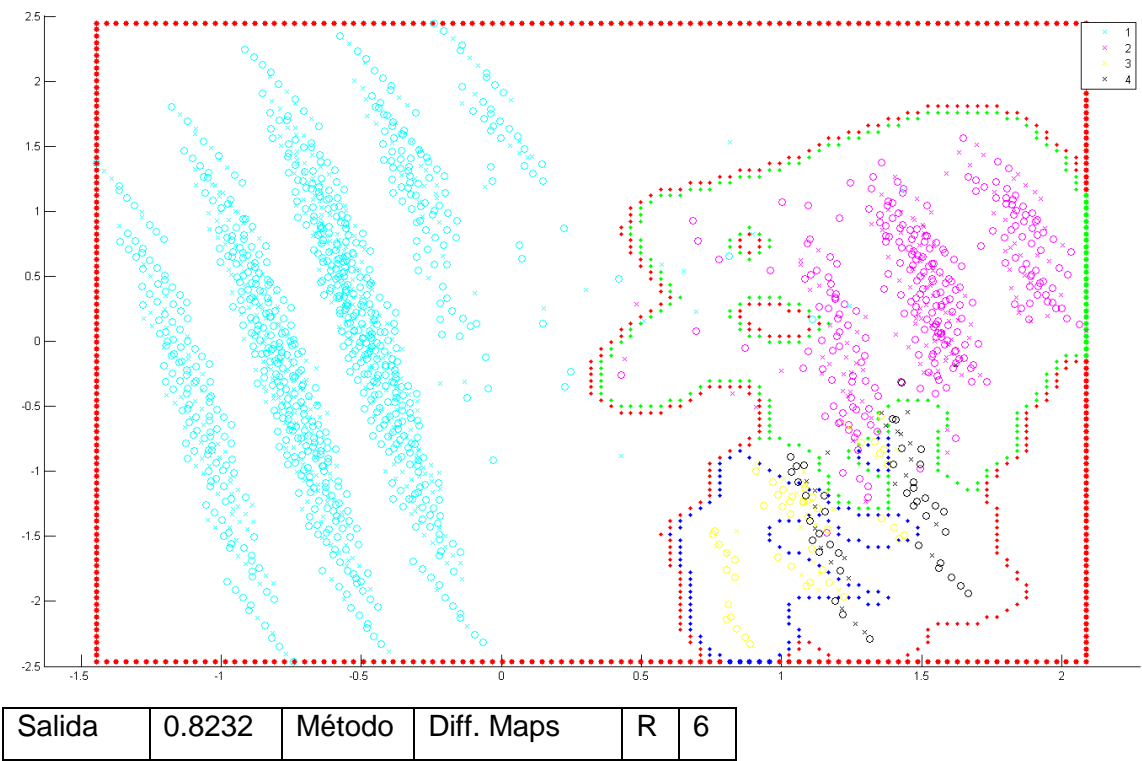
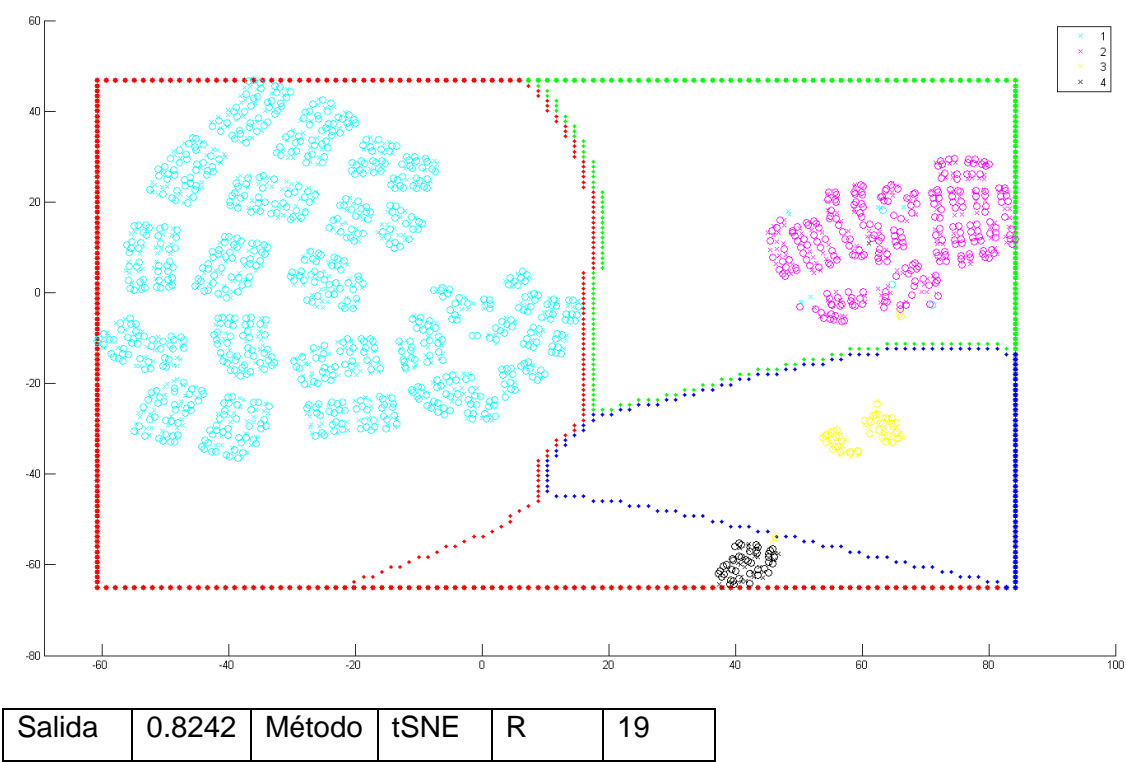
Visualización bidimensional de problemas de clasificación en alta dimensión

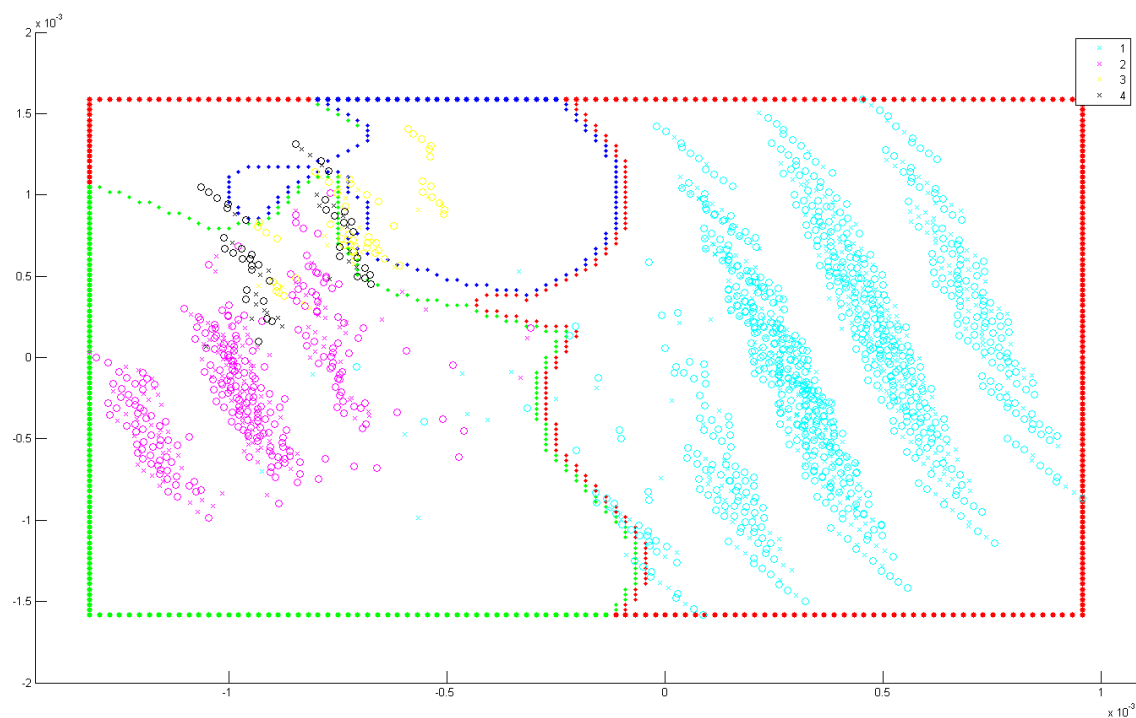




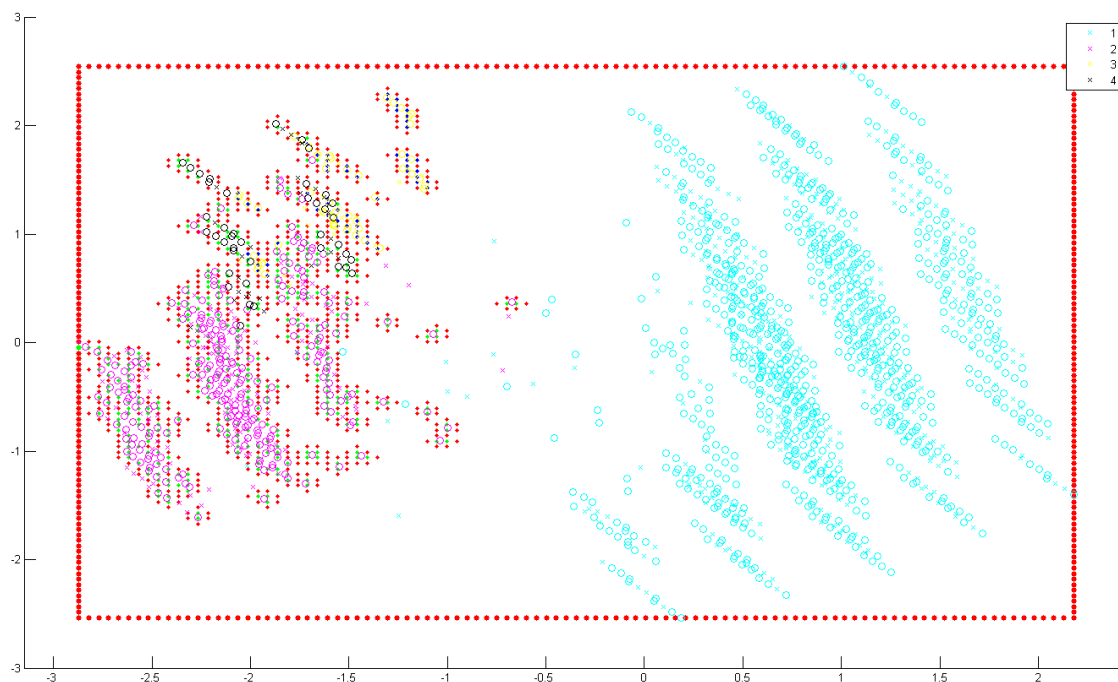


Visualización bidimensional de problemas de clasificación en alta dimensión



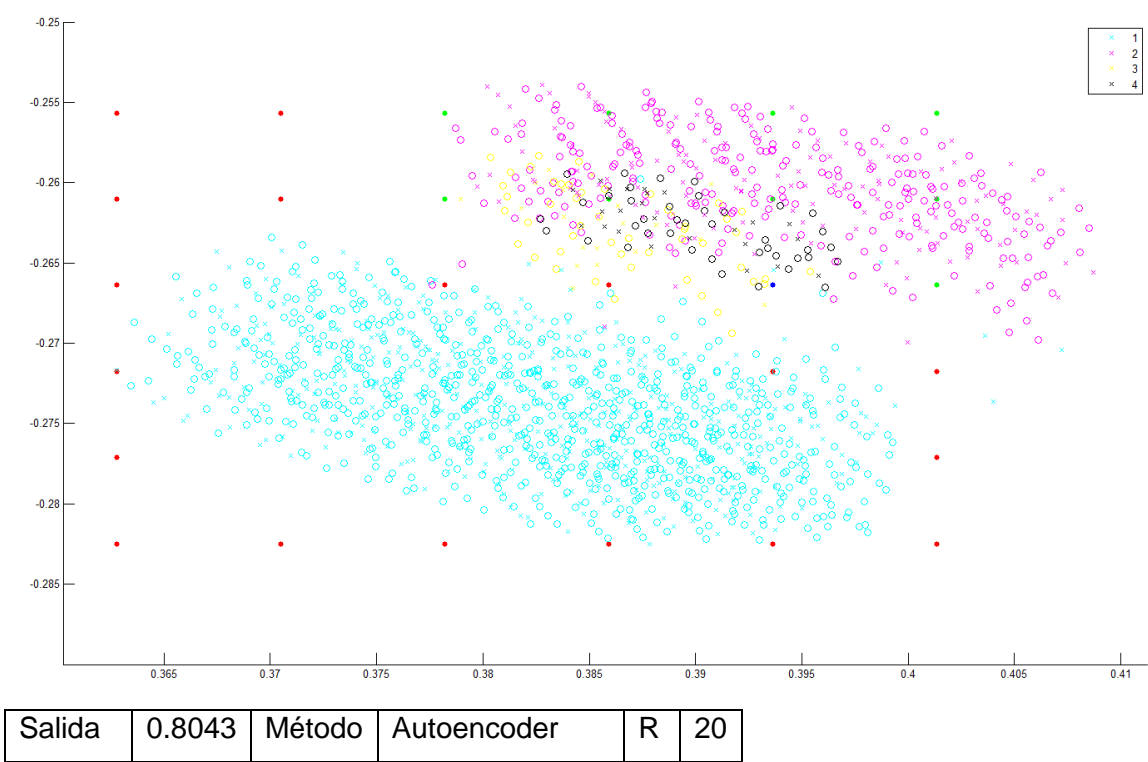
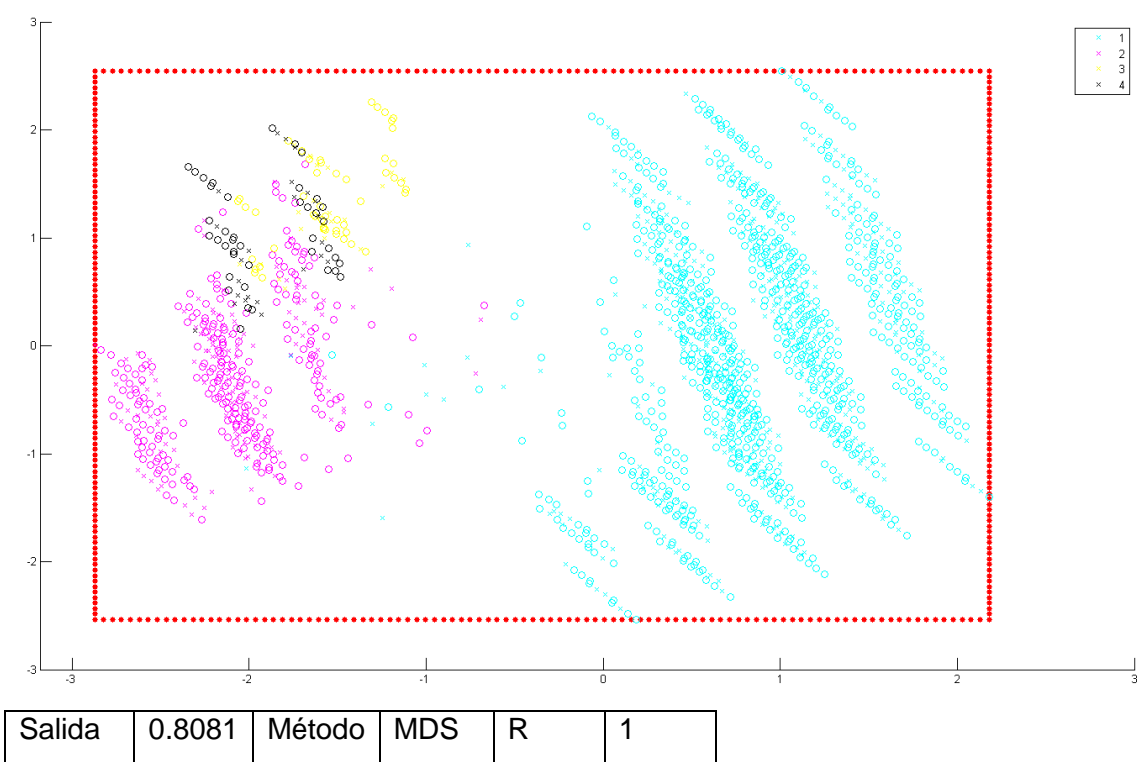


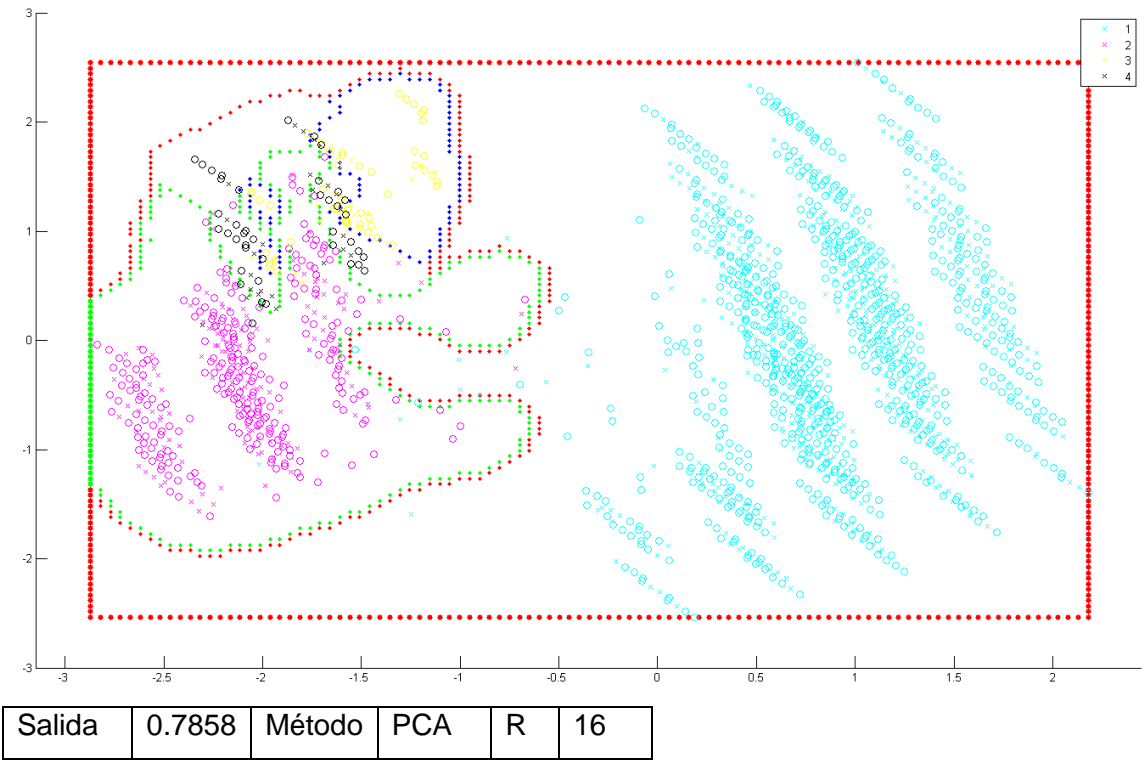
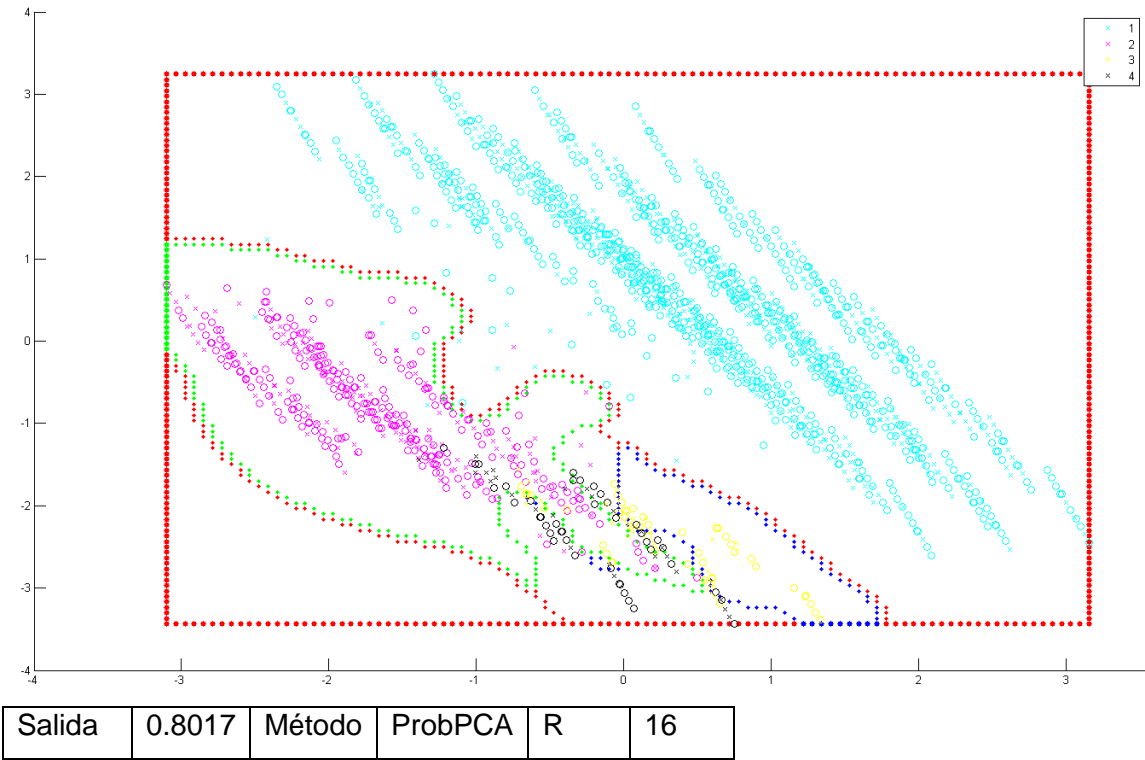
Salida	0.8129	Método	Laplacian	R	7
--------	--------	--------	-----------	---	---



Salida	0.8087	Método	GPLVM	R	1
--------	--------	--------	-------	---	---

Visualización bidimensional de problemas de clasificación en alta dimensión





## 5.2 Resultados numéricos ampliados

Este anexo contiene de forma ampliada los resultados obtenidos para las técnicas de reducción de dimensionalidad. Es decir, vamos a representar el valor de cada uno de los seis parámetros sin normalizar, de esta forma se puede comprobar cuanto ha afectado cada uno de ellos en el resultado final.

Para saber cuanto ha aportado cada dato al resultado Total, se procede a describir el procedimiento.

- Para la columna del porcentaje el único cálculo que hay que realizar es dividir entre 100.
- En la columna de Kendall- $\tau$  hay que sumarle +1 al resultado y posteriormente dividir entre 2
- En la columna de los errores tenemos que tener en cuenta los errores que existían en N dimensiones, estos son los siguientes:
  - Iris: 4 en train y 1 en test
  - Wine: 0 en train y 1 en test
  - Balance: 0 en train y 2 en test
  - Car: 5 en train y 12 en test

Ahora se tomaban las siguientes decisiones. Si no hay errores en N dimensiones y tampoco en 2 Dimensiones la salida es 1. Si no hay errores en N dimensiones y hay algún error en dos dimensiones se da como salida 0. Si existen errores en ND y existen errores coincidentes, se suma el numero de errores en ND el número de errores en 2D y se divide dicha suma entre el doble del número de errores en ND.

Finalmente como existen seis parámetros cada salida se divide entre seis.

### 5.2.1 Conjunto de datos Iris

Autoen	Train			Test			Salida
Factor	%	K.T	Errs.	%	K.T	Errs.	
0.0	93.0000	-0.0836	3	94.0000	-0.1567	0	0.6875
0.1	91.0000	-0.1369	2	92.0000	-0.1771	0	0.6538
0.2	93.0000	-0.1081	3	96.0000	-0.1369	0	0.6904
0.3	92.0000	-0.1281	3	94.0000	-0.1443	0	0.6831
0.4	93.0000	-0.1025	3	94.0000	-0.0708	0	0.6931
0.5	93.0000	-0.0844	3	96.0000	-0.1787	0	0.6889
0.6	94.0000	-0.1275	3	98.0000	-0.1685	0	0.6912
0.7	94.0000	-0.0973	3	96.0000	-0.2072	0	0.6871
0.8	94.0000	-0.0992	1	96.0000	-0.1626	1	0.7323
0.9	94.0000	-0.0872	3	96.0000	-0.1804	0	0.6902
1.0	94.0000	-0.0986	3	98.0000	-0.1622	0	0.6941
1.1	94.0000	-0.0754	3	98.0000	-0.1754	0	0.6949
1.2	95.0000	-0.1063	3	100.0000	-0.0598	0	0.7070
1.3	99.0000	-0.0640	0	96.0000	-0.1186	1	0.7265
1.4	95.0000	-0.1337	3	98.0000	-0.1781	0	0.6915
1.5	100.0000	-0.1226	0	94.0000	-0.1669	1	0.7159
1.6	92.0000	-0.0924	0	92.0000	-0.1100	1	0.7065
1.7	96.0000	-0.0920	2	98.0000	-0.1230	0	0.6804
1.8	95.0000	-0.0764	3	100.0000	-0.1459	0	0.7023
1.9	97.0000	-0.0813	0	96.0000	-0.1943	1	0.7154
2.0	95.0000	-0.1060	3	100.0000	-0.1683	0	0.6980

Diffusion Maps	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	91.0000	-0.1170	1	92.0000	-0.1109	0	0.6402
0.1	91.0000	-0.1170	3	90.0000	-0.1109	0	0.6785
0.2	91.0000	-0.1164	2	92.0000	-0.1109	0	0.6611
0.3	91.0000	-0.1140	1	94.0000	-0.1084	0	0.6440
0.4	92.0000	-0.1135	1	92.0000	-0.1038	0	0.6427
0.5	92.0000	-0.1141	1	94.0000	-0.1013	0	0.6462
0.6	92.0000	-0.1145	1	92.0000	-0.0992	0	0.6430
0.7	92.0000	-0.1134	1	94.0000	-0.0904	0	0.6472
0.8	92.0000	-0.1120	2	92.0000	-0.0825	0	0.6655
0.9	92.0000	-0.1102	1	98.0000	-0.0801	0	0.6550
1.0	91.0000	-0.1150	2	98.0000	-0.0889	0	0.6730
1.1	91.0000	-0.1124	2	98.0000	-0.0962	0	0.6726
1.2	87.0000	-0.1149	0	86.0000	-0.0984	1	0.6872
1.3	92.0000	-0.1165	3	88.0000	-0.0867	0	0.6789
1.4	94.0000	-0.1185	3	94.0000	-0.0924	0	0.6916
1.5	95.0000	-0.1168	2	94.0000	-0.0869	0	0.6730
1.6	97.0000	-0.1158	0	92.0000	-0.1118	1	0.7127
1.7	97.0000	-0.1106	0	94.0000	-0.1042	1	0.7171
1.8	96.0000	-0.1089	0	94.0000	-0.0891	1	0.7168
1.9	88.0000	-0.1172	4	88.0000	-0.0947	0	0.6923
2.0	94.0000	-0.1119	0	96.0000	-0.0914	1	0.7164



GPLVM	Train			Test			Salida
Factor	%	K.T	Errs.	%	K.T	Errs.	
0.0	91.0000	-0.1261	3	92.0000	-0.1416	0	0.6785
0.1	92.0000	-0.1261	2	92.0000	-0.1416	0	0.6594
0.2	92.0000	-0.1266	2	92.0000	-0.1392	0	0.6595
0.3	93.0000	-0.1267	1	92.0000	-0.1423	0	0.6401
0.4	92.0000	-0.1278	1	92.0000	-0.1470	0	0.6379
0.5	91.0000	-0.1278	1	94.0000	-0.1445	0	0.6398
0.6	91.0000	-0.1296	1	92.0000	-0.1445	1	0.7197
0.7	93.0000	-0.1298	2	92.0000	-0.1413	0	0.6607
0.8	93.0000	-0.1323	0	94.0000	-0.1382	0	0.6225
0.9	91.0000	-0.1291	3	96.0000	-0.1447	0	0.6847
1.0	87.0000	-0.1313	0	88.0000	-0.1491	1	0.6850
1.1	91.0000	-0.1286	3	96.0000	-0.1520	0	0.6841
1.2	93.0000	-0.1324	3	96.0000	-0.1520	0	0.6871
1.3	92.0000	-0.1308	3	96.0000	-0.1510	0	0.6857
1.4	94.0000	-0.1341	3	96.0000	-0.1424	0	0.6895
1.5	95.0000	-0.1286	2	96.0000	-0.1378	0	0.6711
1.6	96.0000	-0.1273	2	94.0000	-0.1369	1	0.7530
1.7	97.0000	-0.1256	0	96.0000	-0.1316	1	0.7169
1.8	96.0000	-0.1237	1	98.0000	-0.1229	1	0.7403
1.9	97.0000	-0.1156	0	96.0000	-0.1356	1	0.7174
2.0	95.0000	-0.1166	3	100.0000	-0.1502	0	0.6986

KernelPCA	Train			Test			Salida
Factor	%	K.T	Errs.	%	K.T	Errs.	
0.0	86.0000	0.0137	1	80.0000	0.0190	1	0.7169
0.1	90.0000	0.0136	1	88.0000	0.0211	0	0.6537
0.2	94.0000	0.0162	1	90.0000	0.0211	0	0.6639
0.3	92.0000	0.0187	1	88.0000	0.0190	0	0.6573
0.4	93.0000	0.0238	1	90.0000	0.0165	0	0.6625
0.5	92.0000	0.0286	2	88.0000	0.0211	0	0.6791
0.6	96.0000	0.0299	1	94.0000	0.0170	0	0.6747
0.7	90.0000	0.0312	1	84.0000	0.0086	1	0.7308
0.8	92.0000	0.0364	1	90.0000	0.0008	1	0.7439
0.9	94.0000	0.0383	1	94.0000	-0.0058	1	0.7535
1.0	93.0000	0.0472	1	94.0000	0.0044	1	0.7535
1.1	94.0000	0.0541	1	96.0000	0.0046	1	0.7591
1.2	95.0000	0.0570	3	96.0000	0.0012	1	0.8023
1.3	97.0000	0.0596	0	96.0000	-0.0083	1	0.7426
1.4	100.0000	0.0624	0	96.0000	-0.0276	1	0.7462
1.5	100.0000	0.0630	0	98.0000	-0.0383	1	0.7487
1.6	99.0000	0.0714	1	98.0000	-0.0425	1	0.7682
1.7	99.0000	0.0724	1	98.0000	-0.0384	1	0.7687
1.8	99.0000	0.0723	1	98.0000	-0.0384	1	0.7687
1.9	99.0000	0.0683	1	98.0000	-0.0396	1	0.7682
2.0	98.0000	0.0679	2	96.0000	-0.0523	1	0.7830

Laplacian	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	90.0000	-0.0961	3	94.0000	-0.1245	0	0.6841
0.1	89.0000	-0.0967	3	94.0000	-0.1213	0	0.6827
0.2	91.0000	-0.0974	3	94.0000	-0.1223	0	0.6859
0.3	90.0000	-0.0916	1	90.0000	-0.1145	1	0.7203
0.4	89.0000	-0.0918	3	94.0000	-0.1152	0	0.6836
0.5	91.0000	-0.0914	3	94.0000	-0.1174	0	0.6868
0.6	90.0000	-0.0938	3	94.0000	-0.1083	0	0.6857
0.7	91.0000	-0.0974	2	92.0000	-0.1061	0	0.6630
0.8	91.0000	-0.0984	2	94.0000	-0.0964	0	0.6671
0.9	92.0000	-0.0936	3	92.0000	-0.0953	0	0.6868
1.0	93.0000	-0.0977	0	98.0000	-0.1028	0	0.6350
1.1	92.0000	-0.0962	2	94.0000	-0.0982	0	0.6688
1.2	92.0000	-0.0988	2	92.0000	-0.0967	1	0.7487
1.3	96.0000	-0.1012	1	88.0000	-0.0972	0	0.6443
1.4	93.0000	-0.1014	1	92.0000	-0.0969	1	0.7293
1.5	95.0000	-0.0988	2	92.0000	-0.0925	1	0.7541
1.6	96.0000	-0.0990	1	94.0000	-0.1086	1	0.7369
1.7	96.0000	-0.0955	1	94.0000	-0.0981	1	0.7380
1.8	98.0000	-0.0914	0	94.0000	-0.0996	1	0.7207
1.9	97.0000	-0.0885	1	96.0000	-0.0915	1	0.7442
2.0	95.0000	-0.0934	3	96.0000	-0.0965	0	0.6983

LDA	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	100.0000	-0.0580	0	94.0000	-0.1491	1	0.7227
0.1	91.0000	-0.0492	2	94.0000	-0.0423	0	0.6757
0.2	32.0000	-0.0750	4	36.0000	-0.1202	1	0.5971
0.3	96.0000	-0.0431	4	98.0000	-0.0660	1	0.8142
0.4	97.0000	-0.0493	3	98.0000	-0.1243	1	0.7897
0.5	91.0000	-0.0492	2	96.0000	-0.0423	0	0.6790
0.6	97.0000	-0.0466	3	98.0000	-0.0503	1	0.7961
0.7	92.0000	-0.0492	1	86.0000	-0.0423	1	0.7265
0.8	98.0000	-0.0512	2	98.0000	-0.1068	1	0.7718
0.9	96.0000	-0.0370	4	98.0000	-0.0758	1	0.8139
1.0	96.0000	-0.0393	4	98.0000	-0.0799	1	0.8134
1.1	92.0000	-0.0492	1	86.0000	-0.0423	1	0.7265
1.2	91.0000	-0.0492	2	96.0000	-0.0423	0	0.6790
1.3	96.0000	-0.0370	4	98.0000	-0.0758	1	0.8139
1.4	92.0000	-0.0492	1	86.0000	-0.0423	1	0.7265
1.5	96.0000	-0.0370	4	98.0000	-0.0758	1	0.8139
1.6	96.0000	-0.0474	4	98.0000	-0.0479	1	0.8154
1.7	98.0000	-0.0400	2	98.0000	-0.0988	1	0.7734
1.8	96.0000	-0.0370	4	98.0000	-0.0758	1	0.8139
1.9	96.0000	-0.0474	4	98.0000	-0.0479	1	0.8154
2.0	91.0000	-0.0492	2	92.0000	-0.0423	0	0.6724

LPP	Train			Test			
R	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	96.0000	-0.1241	2	98.0000	-0.1472	0	0.6757
0.1	94.0000	-0.0554	1	90.0000	-0.1029	1	0.7310
0.2	89.0000	-0.0554	0	86.0000	-0.0985	1	0.6955
0.3	89.0000	-0.0659	0	84.0000	-0.1098	1	0.6904
0.4	94.0000	-0.0646	2	92.0000	-0.1118	1	0.7536
0.5	91.0000	-0.0646	0	88.0000	-0.1161	1	0.6999
0.6	96.0000	-0.0513	1	98.0000	-0.1107	1	0.7473
0.7	93.0000	-0.0564	0	92.0000	-0.1058	1	0.7115
0.8	93.0000	-0.0577	0	90.0000	-0.1101	1	0.7077
0.9	94.0000	-0.0631	0	90.0000	-0.1058	1	0.7093
1.0	94.0000	-0.0618	0	90.0000	-0.1052	1	0.7094
1.1	94.0000	-0.0619	0	90.0000	-0.1101	1	0.7090
1.2	97.0000	-0.0606	1	98.0000	-0.1101	1	0.7483
1.3	95.0000	-0.0581	2	98.0000	-0.1150	1	0.7656
1.4	97.0000	-0.0606	1	98.0000	-0.1058	1	0.7486
1.5	96.0000	-0.0579	2	98.0000	-0.1101	1	0.7677
1.6	97.0000	-0.0620	2	98.0000	-0.0998	1	0.7698
1.7	95.0000	-0.0569	2	98.0000	-0.1047	1	0.7665
1.8	98.0000	-0.0639	0	92.0000	-0.1003	0	0.6363
1.9	94.0000	-0.0638	0	90.0000	-0.0954	1	0.7101
2.0	95.0000	-0.0724	0	90.0000	-0.0956	1	0.7110

MCML	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	96.0000	-0.0495	3	100.0000	-0.1483	0	0.7060
0.1	96.0000	-0.0583	4	96.0000	-0.1519	1	0.8025
0.2	99.0000	-0.0684	1	98.0000	-0.1468	1	0.7479
0.3	98.0000	-0.0670	2	98.0000	-0.1807	1	0.7644
0.4	98.0000	-0.0676	2	98.0000	-0.1596	1	0.7661
0.5	100.0000	-0.0598	0	92.0000	-0.1449	1	0.7196
0.6	98.0000	-0.0688	2	98.0000	-0.1504	1	0.7667
0.7	98.0000	-0.0871	2	98.0000	-0.1197	1	0.7678
0.8	98.0000	-0.0852	2	98.0000	-0.0991	1	0.7696
0.9	98.0000	-0.0884	2	98.0000	-0.0817	1	0.7708
1.0	98.0000	-0.0868	2	98.0000	-0.0682	1	0.7721
1.1	98.0000	-0.0838	2	98.0000	-0.0502	1	0.7738
1.2	98.0000	-0.0831	2	98.0000	-0.0426	1	0.7745
1.3	98.0000	-0.0866	2	98.0000	-0.0338	1	0.7750
1.4	98.0000	-0.0945	2	98.0000	-0.0494	1	0.7730
1.5	98.0000	-0.0883	2	98.0000	-0.0559	1	0.7730
1.6	98.0000	-0.0890	2	98.0000	-0.0576	1	0.7728
1.7	98.0000	-0.0904	2	98.0000	-0.0761	1	0.7711
1.8	98.0000	-0.0852	2	98.0000	-0.0730	1	0.7718
1.9	98.0000	-0.0899	2	98.0000	-0.0832	1	0.7706
2.0	94.0000	-0.0713	0	94.0000	-0.0533	1	0.7196

MDS	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	91.0000	-0.1261	3	92.0000	-0.1416	0	0.6785
0.1	92.0000	-0.1261	2	92.0000	-0.1416	0	0.6594
0.2	92.0000	-0.1266	2	92.0000	-0.1392	0	0.6595
0.3	93.0000	-0.1267	1	92.0000	-0.1423	0	0.6401
0.4	92.0000	-0.1278	1	92.0000	-0.1470	0	0.6379
0.5	91.0000	-0.1278	1	94.0000	-0.1445	0	0.6398
0.6	91.0000	-0.1296	1	92.0000	-0.1445	1	0.7197
0.7	93.0000	-0.1298	2	92.0000	-0.1413	0	0.6607
0.8	93.0000	-0.1323	0	94.0000	-0.1382	0	0.6225
0.9	91.0000	-0.1291	3	96.0000	-0.1447	0	0.6847
1.0	87.0000	-0.1313	0	88.0000	-0.1491	1	0.6850
1.1	91.0000	-0.1286	3	96.0000	-0.1520	0	0.6841
1.2	93.0000	-0.1324	3	96.0000	-0.1520	0	0.6871
1.3	92.0000	-0.1308	3	96.0000	-0.1510	0	0.6857
1.4	94.0000	-0.1341	3	96.0000	-0.1424	0	0.6895
1.5	95.0000	-0.1286	2	96.0000	-0.1378	0	0.6711
1.6	96.0000	-0.1273	2	94.0000	-0.1369	1	0.7530
1.7	97.0000	-0.1256	0	96.0000	-0.1316	1	0.7169
1.8	96.0000	-0.1237	1	98.0000	-0.1229	1	0.7403
1.9	97.0000	-0.1156	0	96.0000	-0.1356	1	0.7174
2.0	95.0000	-0.1166	3	100.0000	-0.1502	0	0.6986

NCA	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	100.0000	-0.0471	0	98.0000	-0.1270	1	0.7322
0.1	100.0000	-0.0492	0	98.0000	-0.1249	1	0.7322
0.2	100.0000	-0.0181	0	98.0000	-0.1364	0	0.6505
0.3	100.0000	-0.0361	0	98.0000	-0.1124	0	0.6510
0.4	97.0000	-0.0619	3	100.0000	-0.1917	0	0.7030
0.5	99.0000	-0.0568	1	98.0000	-0.1662	1	0.7473
0.6	99.0000	-0.0550	1	98.0000	-0.1562	1	0.7482
0.7	99.0000	-0.0570	1	98.0000	-0.1640	1	0.7474
0.8	99.0000	-0.0124	1	98.0000	-0.1701	1	0.7506
0.9	100.0000	-0.0383	0	96.0000	-0.2310	1	0.7209
1.0	100.0000	-0.0765	0	100.0000	-0.1337	0	0.6492
1.1	99.0000	-0.0846	1	98.0000	-0.1878	1	0.7431
1.2	99.0000	-0.0751	1	98.0000	-0.2045	1	0.7425
1.3	100.0000	-0.1286	0	98.0000	-0.1412	0	0.6408
1.4	97.0000	-0.0920	1	98.0000	-0.2613	1	0.7331
1.5	99.0000	-0.0920	1	98.0000	-0.1496	1	0.7457
1.6	100.0000	-0.0811	0	96.0000	-0.2154	1	0.7186
1.7	99.0000	-0.0946	1	98.0000	-0.1537	1	0.7451
1.8	99.0000	-0.1086	1	98.0000	-0.1318	1	0.7458
1.9	99.0000	-0.1193	1	98.0000	-0.1227	1	0.7457
2.0	99.0000	-0.1474	1	98.0000	-0.0684	1	0.7479



PCA	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	91.0000	-0.1261	3	92.0000	-0.1416	0	0.6785
0.1	92.0000	-0.1261	2	92.0000	-0.1416	0	0.6594
0.2	92.0000	-0.1266	2	92.0000	-0.1392	0	0.6595
0.3	93.0000	-0.1267	1	92.0000	-0.1423	0	0.6401
0.4	92.0000	-0.1278	1	92.0000	-0.1470	0	0.6379
0.5	91.0000	-0.1278	1	94.0000	-0.1445	0	0.6398
0.6	91.0000	-0.1296	1	92.0000	-0.1445	1	0.7197
0.7	93.0000	-0.1298	2	92.0000	-0.1413	0	0.6607
0.8	93.0000	-0.1323	0	94.0000	-0.1382	0	0.6225
0.9	91.0000	-0.1291	3	96.0000	-0.1447	0	0.6847
1.0	87.0000	-0.1313	0	88.0000	-0.1491	1	0.6850
1.1	91.0000	-0.1286	3	96.0000	-0.1520	0	0.6841
1.2	93.0000	-0.1324	3	96.0000	-0.1520	0	0.6871
1.3	92.0000	-0.1308	3	96.0000	-0.1510	0	0.6857
1.4	94.0000	-0.1341	3	96.0000	-0.1424	0	0.6895
1.5	95.0000	-0.1286	2	96.0000	-0.1378	0	0.6711
1.6	96.0000	-0.1273	2	94.0000	-0.1369	1	0.7530
1.7	97.0000	-0.1256	0	96.0000	-0.1316	1	0.7169
1.8	96.0000	-0.1237	1	98.0000	-0.1229	1	0.7403
1.9	97.0000	-0.1156	0	96.0000	-0.1356	1	0.7174
2.0	95.0000	-0.1166	3	100.0000	-0.1502	0	0.6986

ProbPCA	Train			Test			Salida
Factor	%	K.T	Errs.	%	K.T	Errs.	
0.0	91.0000	-0.1163	2	96.0000	-0.1077	0	0.6680
0.1	91.0000	-0.1133	2	92.0000	-0.1050	0	0.6618
0.2	92.0000	-0.1108	2	96.0000	-0.0933	0	0.6713
0.3	91.0000	-0.1121	1	94.0000	-0.1099	0	0.6440
0.4	90.0000	-0.1123	3	94.0000	-0.0969	0	0.6851
0.5	91.0000	-0.1156	1	90.0000	-0.0901	1	0.7220
0.6	93.0000	-0.1111	0	94.0000	-0.0879	0	0.6284
0.7	93.0000	-0.1102	0	92.0000	-0.0986	1	0.7076
0.8	90.0000	-0.1092	0	90.0000	-0.0933	1	0.6998
0.9	91.0000	-0.1102	0	88.0000	-0.0823	1	0.6990
1.0	92.0000	-0.1121	1	92.0000	-0.0879	0	0.6442
1.1	92.0000	-0.1191	2	98.0000	-0.0957	0	0.6738
1.2	92.0000	-0.1167	2	98.0000	-0.0913	0	0.6743
1.3	87.0000	-0.1140	0	82.0000	-0.0857	1	0.6817
1.4	97.0000	-0.1181	0	88.0000	-0.0779	0	0.6253
1.5	94.0000	-0.1216	3	94.0000	-0.0717	0	0.6931
1.6	94.0000	-0.1208	3	94.0000	-0.0703	0	0.6932
1.7	98.0000	-0.1125	0	92.0000	-0.0903	1	0.7164
1.8	97.0000	-0.1098	1	96.0000	-0.0898	1	0.7425
1.9	97.0000	-0.1052	1	96.0000	-0.0824	1	0.7435
2.0	95.0000	-0.1107	3	96.0000	-0.0911	0	0.6974

SNE	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	88.0000	-0.0345	1	84.0000	-0.0375	0	0.6348
0.1	88.0000	-0.0149	1	84.0000	0.0302	0	0.6421
0.2	88.0000	-0.0060	0	88.0000	0.0276	1	0.7118
0.3	94.0000	-0.0571	3	98.0000	-0.0060	1	0.7939
0.4	94.0000	-0.0485	3	96.0000	-0.0549	1	0.7872
0.5	95.0000	-0.0580	3	94.0000	-0.0683	1	0.7836
0.6	95.0000	-0.0395	3	98.0000	-0.0538	1	0.7931
0.7	94.0000	-0.0407	4	96.0000	-0.0268	1	0.8110
0.8	95.0000	-0.0524	3	98.0000	-0.0413	1	0.7930
0.9	96.0000	-0.1300	2	98.0000	-0.1976	1	0.7544
1.0	95.0000	-0.0672	3	98.0000	-0.0644	1	0.7899
1.1	96.0000	-0.1055	3	98.0000	-0.0869	1	0.7865
1.2	95.0000	-0.0393	4	96.0000	-0.0560	1	0.8104
1.3	95.0000	-0.0448	4	98.0000	-0.0879	1	0.8106
1.4	96.0000	-0.1051	3	98.0000	-0.1362	1	0.7824
1.5	96.0000	-0.0396	4	98.0000	-0.0292	1	0.8176
1.6	97.0000	-0.0719	2	98.0000	-0.0944	1	0.7695
1.7	96.0000	-0.1280	3	98.0000	-0.1848	1	0.7764
1.8	96.0000	-0.0285	3	98.0000	-0.0238	1	0.7981
1.9	96.0000	-0.1287	3	98.0000	-0.2354	1	0.7722
2.0	96.0000	-0.0762	4	98.0000	-0.0647	1	0.8116

SymSNE	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	36.0000	-0.0081	1	32.0000	-0.0321	1	0.5475
0.1	32.0000	0.0141	0	28.0000	0.0649	1	0.5232
0.2	60.0000	0.0667	1	24.0000	-0.0500	0	0.4956
0.3	1.0000	0.0183	4	30.0000	-0.0282	1	0.5508
0.4	2.1100	0.0271	4	24.0000	0.0693	0	0.4682
0.5	30.0000	0.0513	0	26.0000	0.1439	1	0.5263
0.6	42.0000	-0.0404	0	34.0000	-0.1968	1	0.5236
0.7	16.0000	0.0229	1	26.0000	0.0417	1	0.5129
0.8	2.0800	-0.0087	4	24.0000	-0.1275	1	0.5321
0.9	43.0000	-0.0218	4	52.0000	-0.0303	1	0.6540
1.0	27.0000	-0.0740	4	32.0000	-0.0915	0	0.5012
1.1	34.0000	0.0356	1	34.0000	0.0526	1	0.5582
1.2	99.0000	-0.0479	0	30.0000	-0.0573	0	0.5396
1.3	65.0000	-0.0488	0	32.0000	-0.0391	1	0.5710
1.4	36.0000	-0.0082	1	42.0000	0.0948	1	0.5747
1.5	42.0000	0.0276	0	30.0000	0.1662	1	0.5528
1.6	52.0000	-0.1347	1	42.0000	0.0361	1	0.5859
1.7	43.0000	-0.0290	3	42.0000	-0.0825	0	0.5282
1.8	45.0000	-0.0224	2	26.0000	0.0502	1	0.5790
1.9	35.0000	-0.1568	0	30.0000	-0.1440	1	0.4999
2.0	29.0000	0.0674	2	30.0000	0.0730	1	0.5684

tSNE	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	97.0000	-0.0569	1	100.0000	-0.0904	0	0.6702
0.1	97.0000	-0.0540	2	98.0000	-0.0632	0	0.6902
0.2	94.0000	-0.0657	3	98.0000	-0.0966	1	0.7856
0.3	93.0000	-0.0441	4	98.0000	-0.0599	1	0.8097
0.4	98.0000	-0.0467	0	98.0000	-0.0777	1	0.7330
0.5	97.0000	-0.0485	1	98.0000	-0.0948	1	0.7506
0.6	95.0000	-0.0270	2	100.0000	-0.0641	0	0.6924
0.7	95.0000	-0.0427	2	98.0000	-0.0842	1	0.7694
0.8	96.0000	-0.0474	2	98.0000	-0.0641	1	0.7724
0.9	95.0000	-0.0315	4	98.0000	-0.0531	1	0.8146
1.0	97.0000	-0.0385	3	100.0000	-0.0607	0	0.7159
1.1	96.0000	-0.0597	4	98.0000	-0.0953	1	0.8104
1.2	96.0000	-0.0553	4	98.0000	-0.0795	1	0.8121
1.3	96.0000	-0.1038	4	98.0000	-0.1279	1	0.8040
1.4	97.0000	-0.0279	3	98.0000	-0.0670	1	0.7963
1.5	96.0000	-0.0606	4	98.0000	-0.1067	1	0.8094
1.6	96.0000	-0.0579	4	98.0000	-0.0670	1	0.8129
1.7	96.0000	-0.0588	4	98.0000	-0.0615	1	0.8133
1.8	96.0000	-0.0510	4	98.0000	-0.1390	1.	0.8075
1.9	96.0000	-0.0603	4	98.0000	-0.0474	1	0.8144
2.0	96.0000	-0.0595	4	100.0000	-0.0594	0	0.7334

**5.2.2 Conjunto de datos Wine**

Autoencoder	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	90.7563	0.0656	0	88.1356	-0.0995	0	0.7120
0.1	89.0756	0.0674	0	86.4407	-0.1031	0	0.7062
0.2	97.4790	0.0373	0	93.2203	-0.1229	0	0.7274
0.3	98.3193	-0.0007	0	91.5254	-0.0502	1	0.8122
0.4	97.4790	0.0052	0	93.2203	-0.1399	0	0.7233
0.5	99.1597	0.0295	0	96.6102	-0.0970	0	0.7373
0.6	96.6387	0.0110	0	89.8305	-0.0319	1	0.8090
0.7	96.6387	0.0503	0	89.8305	-0.1699	1	0.8008
0.8	98.3193	0.0652	0	93.2203	-0.1062	0	0.7325
0.9	95.7983	0.0633	0	93.2203	-0.0647	1	0.8149
1.0	97.4790	0.0533	0	93.2203	-0.0860	0	0.7318
1.1	100.0000	0.0243	0	98.3051	-0.0783	1	0.8260
1.2	98.3193	0.0360	0	96.6102	-0.1210	0	0.7345
1.3	99.1597	0.0683	0	96.6102	-0.0992	0	0.7404
1.4	100.0000	0.0230	0	93.2203	-0.0946	1	0.8161
1.5	99.1597	-0.0138	0	94.9153	-0.1557	0	0.7260
1.6	100.0000	0.0587	0	94.9153	-0.0531	1	0.8253
1.7	98.3193	0.0870	0	93.2203	-0.1265	1	0.8159
1.8	100.0000	0.0147	0	93.2203	-0.1313	1	0.8123
1.9	99.1597	0.0275	0	96.6102	-0.1464	1	0.8164
2.0	100.0000	0.0594	0	94.9153	-0.0261	1	0.8276

Diffusion Maps	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	97.4790	0.0349	0	96.6102	-0.1095	0	0.7339
0.1	96.6387	0.0349	0	94.9153	-0.1095	1	0.8130
0.2	96.6387	0.0340	0	93.2203	-0.1095	1	0.8101
0.3	96.6387	0.0348	0	93.2203	-0.1095	1	0.8102
0.4	97.4790	0.0346	0	94.9153	-0.1090	1	0.8145
0.5	97.4790	0.0353	0	94.9153	-0.1079	1	0.8146
0.6	97.4790	0.0344	0	94.9153	-0.1051	1	0.8148
0.7	97.4790	0.0336	0	94.9153	-0.1041	1	0.8148
0.8	97.4790	0.0334	0	94.9153	-0.1084	1	0.8144
0.9	97.4790	0.0339	0	94.9153	-0.1018	1	0.8150
1.0	100.0000	0.0342	0	94.9153	-0.1057	1	0.8189
1.1	99.1597	0.0356	0	94.9153	-0.1029	1	0.8178
1.2	100.0000	0.0355	0	94.9153	-0.1057	1	0.8190
1.3	100.0000	0.0349	0	94.9153	-0.1089	1	0.8187
1.4	100.0000	0.0347	0	94.9153	-0.1172	1	0.8180
1.5	100.0000	0.0371	0	94.9153	-0.1200	1	0.8180
1.6	100.0000	0.0363	0	94.9153	-0.1145	1	0.8183
1.7	100.0000	0.0357	0	94.9153	-0.1100	1	0.8187
1.8	100.0000	0.0355	0	94.9153	-0.1034	1	0.8192
1.9	100.0000	0.0348	0	96.6102	-0.0973	1	0.8225
2.0	100.0000	0.0338	0	96.6102	-0.0918	1	0.8229

GPLVM	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	96.6387	0.0369	0	93.2203	-0.1482	1	0.8072
0.1	96.6387	0.0369	0	93.2203	-0.1482	1	0.8072
0.2	96.6387	0.0366	0	93.2203	-0.1455	1	0.8074
0.3	96.6387	0.0364	0	93.2203	-0.1455	1	0.8073
0.4	96.6387	0.0350	0	93.2203	-0.1455	1	0.8072
0.5	96.6387	0.0360	0	93.2203	-0.1466	1	0.8072
0.6	96.6387	0.0360	0	96.6102	-0.1466	0	0.7295
0.7	97.4790	0.0385	0	94.9153	-0.1466	1	0.8116
0.8	96.6387	0.0403	0	94.9153	-0.1461	1	0.8104
0.9	97.4790	0.0403	0	94.9153	-0.1432	1	0.8121
1.0	97.4790	0.0406	0	94.9153	-0.1421	1	0.8122
1.1	99.1597	0.0405	0	94.9153	-0.1482	1	0.8145
1.2	100.0000	0.0396	0	91.5254	-0.1465	1	0.8103
1.3	100.0000	0.0394	0	91.5254	-0.1448	1	0.8104
1.4	100.0000	0.0372	0	94.9153	-0.1404	1	0.8163
1.5	100.0000	0.0385	0	94.9153	-0.1415	1	0.8163
1.6	100.0000	0.0383	0	91.5254	-0.1447	1	0.8103
1.7	100.0000	0.0384	0	94.9153	-0.1509	1	0.8155
1.8	100.0000	0.0368	0	94.9153	-0.1531	1	0.8152
1.9	100.0000	0.0362	0	94.9153	-0.1541	1	0.8150
2.0	100.0000	0.0353	0	94.9153	-0.1459	1	0.8156



KernelPCA	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	94.9580	0.0457	0	86.4407	-0.0288	1	0.8037
0.1	94.9580	0.0443	0	88.1356	-0.0288	1	0.8064
0.2	94.1176	0.0456	0	84.7458	-0.0277	1	0.7996
0.3	94.1176	0.0478	0	84.7458	-0.0277	1	0.7998
0.4	96.6387	0.0487	0	88.1356	-0.0222	1	0.8102
0.5	96.6387	0.0489	0	88.1356	-0.0239	1	0.8100
0.6	91.5966	0.0473	0	84.7458	-0.0278	1	0.7955
0.7	94.9580	0.0506	0	88.1356	-0.0267	1	0.8072
0.8	95.7983	0.0482	0	86.4407	-0.0267	1	0.8055
0.9	93.2773	0.0493	0	86.4407	-0.0289	1	0.8012
1.0	94.9580	0.0492	0	88.1356	-0.0261	1	0.8071
1.1	95.7983	0.0503	0	91.5254	-0.0228	1	0.8145
1.2	94.9580	0.0473	0	89.8305	-0.0206	1	0.8102
1.3	94.1176	0.0485	0	94.9153	-0.0206	0	0.7340
1.4	94.1176	0.0492	0	94.9153	-0.0184	0	0.7343
1.5	94.1176	0.0512	0	96.6102	-0.0173	0	0.7374
1.6	94.1176	0.0510	0	96.6102	-0.0173	0	0.7374
1.7	94.1176	0.0513	0	96.6102	-0.0162	0	0.7375
1.8	94.1176	0.0510	0	94.9153	-0.0151	1	0.8180
1.9	94.1176	0.0507	0	94.9153	-0.0151	1	0.8180
2.0	94.1176	0.0505	0	94.9153	-0.0151	1	0.8180

Laplacian	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	96.6387	0.0373	0	94.9153	-0.1024	1	0.8138
0.1	96.6387	0.0378	0	94.9153	-0.1024	1	0.8139
0.2	96.6387	0.0383	0	93.2203	-0.1012	1	0.8112
0.3	96.6387	0.0382	0	94.9153	-0.0985	1	0.8142
0.4	97.4790	0.0367	0	94.9153	-0.1018	1	0.8152
0.5	97.4790	0.0367	0	94.9153	-0.1001	1	0.8154
0.6	97.4790	0.0353	0	94.9153	-0.0985	1	0.8154
0.7	97.4790	0.0379	0	94.9153	-0.0985	1	0.8156
0.8	97.4790	0.0380	0	94.9153	-0.1007	1	0.8154
0.9	97.4790	0.0365	0	94.9153	-0.1024	1	0.8152
1.0	98.3193	0.0331	0	94.9153	-0.1012	1	0.8164
1.1	98.3193	0.0320	0	94.9153	-0.0969	1	0.8167
1.2	99.1597	0.0323	0	94.9153	-0.1012	1	0.8177
1.3	100.0000	0.0334	0	94.9153	-0.1001	1	0.8193
1.4	100.0000	0.0349	0	94.9153	-0.1046	1	0.8190
1.5	100.0000	0.0327	0	94.9153	-0.1024	1	0.8190
1.6	100.0000	0.0335	0	94.9153	-0.0997	1	0.8193
1.7	100.0000	0.0327	0	94.9153	-0.1029	1	0.8190
1.8	100.0000	0.0353	0	94.9153	-0.0980	1	0.8196
1.9	100.0000	0.0375	0	94.9153	-0.0985	1	0.8198
2.0	100.0000	0.0351	0	96.6102	-0.0891	1	0.8232

LDA	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	100.0000	-0.0100	0	100.0000	-0.0456	0	0.7454
0.1	100.0000	0.0424	0	84.7458	0.0264	1	0.8136
0.2	100.0000	0.0481	0	84.7458	0.0253	1	0.8140
0.3	100.0000	0.0282	0	86.4407	-0.0285	1	0.8107
0.4	100.0000	0.0481	0	91.5254	0.0259	1	0.8254
0.5	100.0000	0.0352	0	84.7458	0.0387	1	0.8141
0.6	100.0000	0.0282	0	86.4407	-0.0285	1	0.8107
0.7	100.0000	0.0282	0	86.4407	-0.0285	1	0.8107
0.8	100.0000	0.0393	0	84.7458	0.0462	1	0.8150
0.9	100.0000	0.0282	0	86.4407	-0.0285	1	0.8107
1.0	100.0000	0.0352	0	84.7458	0.0387	1	0.8141
1.1	100.0000	0.0282	0	86.4407	-0.0285	1	0.8107
1.2	100.0000	0.0282	0	86.4407	-0.0285	1	0.8107
1.3	100.0000	0.0352	0	84.7458	0.0387	1	0.8141
1.4	100.0000	0.0282	0	86.4407	-0.0285	1	0.8107
1.5	100.0000	0.0352	0	84.7458	0.0387	1	0.8141
1.6	100.0000	0.0387	0	84.7458	0.0409	1	0.8145
1.7	100.0000	0.0282	0	86.4407	-0.0285	1	0.8107
1.8	100.0000	0.0348	0	86.4407	-0.0200	1	0.8120
1.9	100.0000	0.0568	0	84.7458	-0.0355	1	0.8097
2.0	100.0000	0.0352	0	84.7458	0.0387	1	0.8141

LPP	Train			Test			
Factor	%	K.T	Errs.	%	K.T	rrs.	Salida
0.0	100.0000	0.0217	0	96.6102	-0.1003	0	0.7378
0.1	100.0000	0.0218	0	94.9153	-0.0761	0	0.7370
0.2	100.0000	0.0228	0	94.9153	-0.0772	0	0.7370
0.3	100.0000	0.0205	0	96.6102	-0.0761	0	0.7397
0.4	100.0000	0.0176	0	96.6102	-0.0783	0	0.7393
0.5	100.0000	0.0179	0	96.6102	-0.0776	0	0.7394
0.6	100.0000	0.0179	0	96.6102	-0.0676	0	0.7402
0.7	100.0000	0.0215	0	96.6102	-0.0739	0	0.7400
0.8	100.0000	0.0293	0	100.0000	-0.0698	0	0.7466
0.9	100.0000	0.0209	0	100.0000	-0.0616	0	0.7466
1.0	100.0000	0.0262	0	98.3051	-0.0529	1	0.8283
1.1	100.0000	0.0273	0	98.3051	-0.0562	1	0.8281
1.2	100.0000	0.0358	0	96.6102	-0.0425	1	0.8271
1.3	100.0000	0.0265	0	94.9153	-0.0581	1	0.8222
1.4	100.0000	0.0227	0	94.9153	-0.0525	1	0.8224
1.5	100.0000	0.0262	0	94.9153	-0.0507	1	0.8228
1.6	100.0000	0.0125	0	93.2203	-0.0490	1	0.8190
1.7	100.0000	0.0076	0	93.2203	-0.0336	1	0.8199
1.8	100.0000	0.0113	0	91.5254	-0.0579	1	0.8153
1.9	100.0000	0.0162	0	91.5254	0.0104	1	0.8214
2.0	100.0000	0.0171	0	89.8305	0.0018	1	0.8180

MCML	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	94.9580	0.0434	0	91.5254	-0.0874	1	0.8071
0.1	100.0000	-0.0108	0	100.0000	-0.0687	0	0.7434
0.2	100.0000	-0.0123	0	100.0000	-0.0650	0	0.7436
0.3	100.0000	-0.0104	0	98.3051	-0.0638	1	0.8243
0.4	100.0000	-0.0100	0	98.3051	-0.0667	1	0.8241
0.5	100.0000	-0.0079	0	98.3051	-0.0552	1	0.8252
0.6	100.0000	-0.0079	0	98.3051	-0.0487	1	0.8258
0.7	100.0000	-0.0030	0	98.3051	-0.0312	1	0.8277
0.8	100.0000	0.0064	0	98.3051	-0.0190	1	0.8295
0.9	100.0000	0.0064	0	98.3051	-0.0240	1	0.8290
1.0	100.0000	0.0083	0	98.3051	-0.0025	1	0.8310
1.1	100.0000	0.0156	0	98.3051	-0.0112	1	0.8309
1.2	100.0000	0.0324	0	98.3051	-0.0085	1	0.8325
1.3	100.0000	0.0413	0	96.6102	-0.0048	1	0.8307
1.4	100.0000	0.0562	0	96.6102	0.0001	1	0.8324
1.5	100.0000	0.0731	0	94.9153	0.0073	1	0.8316
1.6	100.0000	0.0725	0	93.2203	0.0083	1	0.8288
1.7	100.0000	0.0754	0	93.2203	0.0216	1	0.8301
1.8	100.0000	0.0804	0	93.2203	0.0154	1	0.8300
1.9	100.0000	0.0416	0	94.9153	-0.0031	1	0.8281
2.0	97.4790	0.0459	0	98.3051	0.0137	0	0.7479

MDS	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	96.6387	0.0369	0	93.2203	-0.1482	1	0.8072
0.1	96.6387	0.0369	0	93.2203	-0.1482	1	0.8072
0.2	96.6387	0.0366	0	93.2203	-0.1455	1	0.8074
0.3	96.6387	0.0364	0	93.2203	-0.1455	1	0.8073
0.4	96.6387	0.0350	0	93.2203	-0.1455	1	0.8072
0.5	96.6387	0.0360	0	93.2203	-0.1466	1	0.8072
0.6	96.6387	0.0360	0	96.6102	-0.1466	0	0.7295
0.7	97.4790	0.0385	0	94.9153	-0.1466	1	0.8116
0.8	96.6387	0.0403	0	94.9153	-0.1461	1	0.8104
0.9	97.4790	0.0403	0	94.9153	-0.1432	1	0.8121
1.0	97.4790	0.0406	0	94.9153	-0.1421	1	0.8122
1.1	99.1597	0.0405	0	94.9153	-0.1482	1	0.8145
1.2	100.0000	0.0396	0	91.5254	-0.1465	1	0.8103
1.3	100.0000	0.0394	0	91.5254	-0.1448	1	0.8104
1.4	100.0000	0.0372	0	94.9153	-0.1404	1	0.8163
1.5	100.0000	0.0385	0	94.9153	-0.1415	1	0.8163
1.6	100.0000	0.0383	0	91.5254	-0.1447	1	0.8103
1.7	100.0000	0.0384	0	94.9153	-0.1509	1	0.8155
1.8	100.0000	0.0368	0	94.9153	-0.1531	1	0.8152
1.9	100.0000	0.0362	0	94.9153	-0.1541	1	0.8150
2.0	100.0000	0.0353	0	94.9153	-0.1459	1	0.8156

NCA	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	100.0000	0.0099	0	100.0000	-0.1128	0	0.7414
0.1	100.0000	0.0093	0	96.6102	-0.0940	1	0.8206
0.2	100.0000	-0.0298	0	98.3051	-0.1075	0	0.7357
0.3	100.0000	0.0479	0	100.0000	-0.0773	0	0.7476
0.4	100.0000	0.0351	0	100.0000	-0.0968	0	0.7449
0.5	100.0000	0.0208	0	100.0000	-0.0775	0	0.7453
0.6	100.0000	-0.0047	0	100.0000	-0.1009	0	0.7412
0.7	100.0000	0.0197	0	100.0000	-0.0774	0	0.7452
0.8	100.0000	0.0380	0	98.3051	-0.0710	0	0.7444
0.9	100.0000	0.0198	0	100.0000	-0.0917	0	0.7440
1.0	100.0000	0.0600	0	100.0000	-0.0716	0	0.7490
1.1	100.0000	0.0459	0	100.0000	-0.0867	0	0.7466
1.2	100.0000	0.0134	0	100.0000	-0.0813	0	0.7443
1.3	100.0000	0.0437	0	100.0000	-0.1262	0	0.7431
1.4	100.0000	-0.0051	0	100.0000	-0.0044	0	0.7492
1.5	100.0000	-0.0099	0	100.0000	-0.0351	0	0.7463
1.6	100.0000	-0.0269	0	98.3051	-0.0575	1	0.8235
1.7	100.0000	-0.0233	0	98.3051	0.0265	0	0.7474
1.8	100.0000	0.0620	0	98.3051	-0.0860	0	0.7452
1.9	100.0000	-0.0013	0	100.0000	-0.0082	0	0.7492
2.0	100.0000	-0.0023	0	100.0000	0.0324	0	0.7525

NPE	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	94.9580	0.0540	0	83.0508	-0.0649	1	0.7958
0.1	69.7479	0.0701	0	64.4068	-0.0824	1	0.7226
0.2	71.4286	0.0692	0	64.4068	-0.0922	1	0.7245
0.3	73.1092	0.0632	0	66.1017	-0.1099	1	0.7281
0.4	71.4286	0.0663	0	64.4068	-0.0899	1	0.7244
0.5	70.5882	0.0467	0	64.4068	-0.1067	1	0.7200
0.6	71.4286	0.0429	0	64.4068	-0.1117	1	0.7207
0.7	75.6303	0.0671	0	67.7966	-0.1103	1	0.7354
0.8	75.6303	0.0719	0	67.7966	-0.0790	1	0.7385
0.9	84.8739	0.0725	0	79.6610	-0.0879	1	0.7729
1.0	84.0336	0.0773	0	83.0508	-0.1015	1	0.7765
1.1	94.9580	0.0714	0	88.1356	-0.0697	1	0.8053
1.2	71.4286	-0.0026	0	55.9322	-0.0912	1	0.7045
1.3	71.4286	-0.0031	0	55.9322	-0.0537	1	0.7075
1.4	71.4286	-0.0283	0	55.9322	-0.0177	1	0.7084
1.5	75.6303	0.0077	0	71.1864	-0.0824	1	0.7385
1.6	73.9496	-0.0288	0	55.9322	-0.0827	1	0.7072
1.7	71.4286	-0.0617	0	55.9322	-0.0575	1	0.7023
1.8	71.4286	-0.0557	0	55.9322	-0.0828	1	0.7007
1.9	71.4286	-0.0537	0	55.9322	-0.0749	1	0.7015
2.0	71.4286	-0.0665	0	55.9322	-0.0880	1	0.6994



PCA	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	96.6387	0.0369	0	93.2203	-0.1482	1	0.8072
0.1	96.6387	0.0369	0	93.2203	-0.1482	1	0.8072
0.2	96.6387	0.0366	0	93.2203	-0.1455	1	0.8074
0.3	96.6387	0.0364	0	93.2203	-0.1455	1	0.8073
0.4	96.6387	0.0350	0	93.2203	-0.1455	1	0.8072
0.5	96.6387	0.0360	0	93.2203	-0.1466	1	0.8072
0.6	96.6387	0.0360	0	96.6102	-0.1466	0	0.7295
0.7	97.4790	0.0385	0	94.9153	-0.1466	1	0.8116
0.8	96.6387	0.0403	0	94.9153	-0.1461	1	0.8104
0.9	97.4790	0.0403	0	94.9153	-0.1432	1	0.8121
1.0	97.4790	0.0406	0	94.9153	-0.1421	1	0.8122
1.1	99.1597	0.0405	0	94.9153	-0.1482	1	0.8145
1.2	100.0000	0.0396	0	91.5254	-0.1465	1	0.8103
1.3	100.0000	0.0394	0	91.5254	-0.1448	1	0.8104
1.4	100.0000	0.0372	0	94.9153	-0.1404	1	0.8163
1.5	100.0000	0.0385	0	94.9153	-0.1415	1	0.8163
1.6	100.0000	0.0383	0	91.5254	-0.1447	1	0.8103
1.7	100.0000	0.0384	0	94.9153	-0.1509	1	0.8155
1.8	100.0000	0.0368	0	94.9153	-0.1531	1	0.8152
1.9	100.0000	0.0362	0	94.9153	-0.1541	1	0.8150
2.0	100.0000	0.0353	0	94.9153	-0.1459	1	0.8156

ProbPCA	Train			Test			Salida
Factor	%	K.T	Errs.	%	K.T	Errs.	
0.0	96.6387	0.0299	0	93.2203	-0.0991	1	0.8107
0.1	96.6387	0.0299	0	93.2203	-0.0991	1	0.8107
0.2	96.6387	0.0308	0	93.2203	-0.0991	1	0.8107
0.3	96.6387	0.0310	0	93.2203	-0.1019	1	0.8105
0.4	97.4790	0.0322	0	94.9153	-0.1019	1	0.8149
0.5	97.4790	0.0331	0	94.9153	-0.0997	1	0.8151
0.6	97.4790	0.0339	0	94.9153	-0.0992	1	0.8152
0.7	97.4790	0.0338	0	94.9153	-0.1002	1	0.8151
0.8	97.4790	0.0335	0	94.9153	-0.0980	1	0.8153
0.9	97.4790	0.0342	0	94.9153	-0.0980	1	0.8153
1.0	98.3193	0.0371	0	94.9153	-0.0991	1	0.8169
1.1	98.3193	0.0346	0	94.9153	-0.1024	1	0.8164
1.2	98.3193	0.0339	0	94.9153	-0.0996	1	0.8166
1.3	100.0000	0.0369	0	94.9153	-0.1030	1	0.8193
1.4	100.0000	0.0353	0	94.9153	-0.1036	1	0.8192
1.5	100.0000	0.0342	0	94.9153	-0.1013	1	0.8193
1.6	100.0000	0.0342	0	94.9153	-0.1124	1	0.8183
1.7	100.0000	0.0306	0	94.9153	-0.1145	1	0.8179
1.8	100.0000	0.0330	0	94.9153	-0.1079	1	0.8186
1.9	100.0000	0.0309	0	96.6102	-0.1024	1	0.8217
2.0	100.0000	0.0357	0	96.6102	-0.0985	1	0.8224

Sammon	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	98.3193	-0.0201	0	93.2203	-0.1258	1	0.8071
0.1	99.1597	-0.0182	0	94.9153	-0.1008	1	0.8135
0.2	97.4790	-0.0297	0	86.4407	-0.0985	1	0.7959
0.3	99.1597	-0.0078	0	91.5254	-0.1147	1	0.8076
0.4	100.0000	-0.0215	0	94.9153	-0.0859	1	0.8159
0.5	100.0000	-0.0198	0	96.6102	-0.1130	1	0.8166
0.6	98.3193	-0.0080	0	94.9153	-0.1182	1	0.8115
0.7	99.1597	0.0015	0	96.6102	-0.1082	1	0.8174
0.8	100.0000	-0.0067	0	96.6102	-0.0925	1	0.8194
0.9	100.0000	-0.0371	0	94.9153	-0.0930	1	0.8140
1.0	100.0000	-0.0096	0	93.2203	-0.0892	1	0.8138
1.1	100.0000	-0.0296	0	98.3051	-0.1429	1	0.8161
1.2	100.0000	-0.0135	0	96.6102	-0.1018	1	0.8181
1.3	100.0000	-0.0072	0	96.6102	-0.1169	1	0.8173
1.4	100.0000	-0.0097	0	98.3051	-0.1123	1	0.8203
1.5	100.0000	-0.0141	0	98.3051	-0.0831	1	0.8224
1.6	98.3193	0.0004	0	96.6102	-0.1472	0	0.7293
1.7	100.0000	-0.0209	0	98.3051	-0.1008	1	0.8204
1.8	100.0000	-0.0086	0	98.3051	-0.1030	1	0.8212
1.9	98.3193	-0.0253	0	98.3051	-0.1295	1	0.8148
2.0	100.0000	-0.0088	0	98.3051	-0.0881	1	0.8224

SNE	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	100.0000	0.0373	0	94.9153	-0.1268	1	0.8174
0.1	98.3193	0.0428	0	93.2203	-0.0643	0	0.7341
0.2	96.6387	0.0330	0	94.9153	-0.0337	1	0.8192
0.3	97.4790	0.0388	0	93.2203	0.0069	1	0.8216
0.4	100.0000	0.0294	0	93.2203	-0.1781	1	0.8096
0.5	100.0000	0.0440	0	93.2203	-0.0798	1	0.8191
0.6	100.0000	0.0269	0	93.2203	-0.1242	1	0.8139
0.7	100.0000	0.0434	0	94.9153	-0.0871	1	0.8212
0.8	100.0000	0.0454	0	96.6102	-0.0478	0	0.7442
0.9	100.0000	-0.0052	0	96.6102	-0.1273	1	0.8166
1.0	100.0000	0.0451	0	96.6102	-0.0981	1	0.8233
1.1	100.0000	0.0363	0	96.6102	-0.1212	1	0.8206
1.2	100.0000	0.0405	0	96.6102	-0.0311	1	0.8285
1.3	100.0000	0.0385	0	96.6102	-0.0156	1	0.8296
1.4	100.0000	0.0379	0	96.6102	-0.0127	1	0.8298
1.5	100.0000	0.0310	0	96.6102	-0.0100	1	0.8294
1.6	100.0000	0.0336	0	96.6102	-0.1196	1	0.8205
1.7	100.0000	0.0412	0	96.6102	-0.1187	1	0.8212
1.8	100.0000	0.0418	0	94.9153	-0.0216	1	0.8265
1.9	100.0000	0.0426	0	94.9153	-0.0228	1	0.8265
2.0	100.0000	0.0280	0	96.6102	-0.1134	1	0.8206

SymSNE	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	44.5378	0.1508	0	33.8983	-0.0518	1	0.6390
0.1	43.6975	-0.0739	0	32.2034	-0.0455	0	0.5332
0.2	1.0000	-0.0476	0	30.5085	-0.0057	1	0.5481
0.3	44.5378	0.0309	0	30.5085	0.1272	1	0.6383
0.4	83.1933	0.0104	0	32.2034	0.0491	1	0.6973
0.5	3.3613	-0.0190	0	30.5085	-0.0493	1	0.5508
0.6	49.5798	-0.0025	0	30.5085	0.0585	1	0.6381
0.7	1.0000	-0.0079	0	30.5085	0.0801	1	0.5585
0.8	49.5798	-0.0149	0	28.8136	0.0395	1	0.6327
0.9	1.0000	0.0278	0	30.5085	-0.1145	1	0.5453
1.0	44.5378	0.0006	0	30.5085	-0.0818	1	0.6183
1.1	42.0168	-0.0913	0	30.5085	0.2092	1	0.6307
1.2	60.5042	-0.0945	0	28.8136	-0.0373	1	0.6379
1.3	45.3782	0.0542	0	30.5085	0.0338	1	0.6338
1.4	47.8992	-0.0174	0	33.8983	-0.0319	1	0.6322
1.5	1.3277	-0.0557	0	23.7288	-0.1645	1	0.5234
1.6	70.5882	-0.0674	0	30.5085	0.0414	1	0.6663
1.7	1.0000	0.0675	0	30.5085	0.0393	1	0.5614
1.8	43.6975	-0.0244	0	32.2034	-0.0244	1	0.6224
1.9	100.0000	-0.0992	0	32.2034	0.0556	1	0.7167
2.0	44.5378	-0.0173	0	30.5085	-0.0491	1	0.6195

tSNE	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	96.6387	0.0638	0	96.6102	-0.0551	0	0.7395
0.1	96.6387	0.0557	0	93.2203	-0.0751	0	0.7315
0.2	96.6387	0.0461	0	94.9153	-0.0668	1	0.8175
0.3	99.1597	0.0716	0	94.9153	-0.0520	1	0.8251
0.4	98.3193	0.0551	0	96.6102	-0.0576	0	0.7413
0.5	97.4790	0.0106	0	96.6102	-0.0759	0	0.7347
0.6	97.4790	0.0580	0	96.6102	-0.0868	1	0.8211
0.7	100.0000	0.0153	0	96.6102	-0.0892	1	0.8215
0.8	98.3193	0.0466	0	94.9153	-0.0593	1	0.8210
0.9	97.4790	0.0406	0	94.9153	-0.0726	1	0.8180
1.0	100.0000	0.0408	0	96.6102	-0.0604	1	0.8261
1.1	99.1597	0.0565	0	96.6102	-0.0788	1	0.8244
1.2	100.0000	0.0435	0	96.6102	-0.0276	1	0.8290
1.3	99.1597	0.0589	0	96.6102	-0.0676	1	0.8256
1.4	100.0000	0.0630	0	96.6102	-0.0540	1	0.8284
1.5	100.0000	0.0634	0	96.6102	-0.0517	1	0.8287
1.6	100.0000	0.0473	0	96.6102	-0.0868	1	0.8244
1.7	100.0000	0.0393	0	94.9153	-0.0525	1	0.8238
1.8	100.0000	0.0486	0	96.6102	-0.0463	1	0.8279
1.9	100.0000	0.0362	0	96.6102	-0.0169	1	0.8293
2.0	100.0000	0.0825	0	96.6102	-0.0118	1	0.8336

### 5.2.3 Conjunto de datos Balance

Autoencoder	Train			Test			Sallida
Factor	%	K.T	Errs.	%	K.T	Errs.	
0.0	77.6978	-0.0297	0	59.6154	-0.1433	2	0.7144
0.1	92.0863	-0.0421	0	62.5000	-0.0366	0	0.6678
0.2	60.4317	-0.0051	0	53.3654	-0.1123	2	0.6799
0.3	58.0336	-0.0705	0	57.2115	-0.1203	2	0.6762
0.4	41.9664	-0.0249	0	42.7885	0.0395	1	0.6008
0.5	59.7122	-0.0329	0	55.2885	-0.0608	1	0.6422
0.6	90.4077	-0.0283	0	86.0577	0.1364	0	0.7198
0.7	78.4173	0.0098	0	65.8654	-0.0426	1	0.6961
0.8	98.3213	-0.0934	0	91.8269	-0.1163	0	0.7161
0.9	96.4029	-0.0357	0	93.2692	-0.0266	0	0.7276
1.0	96.1631	-0.0412	0	89.4231	-0.0435	0	0.7189
1.1	97.1223	-0.0236	0	93.7500	0.0306	0	0.7354
1.2	89.9281	-0.0292	0	87.9808	-0.1177	0	0.7009
1.3	98.0815	-0.0444	0	98.0769	-0.0232	0	0.7380
1.4	98.8010	-0.0308	0	96.6346	0.0587	0	0.7447
1.5	94.0048	-0.0435	0	91.8269	-0.0740	1	0.7583
1.6	93.5252	-0.0052	0	93.7500	0.0340	1	0.7729
1.7	98.5612	-0.0250	0	98.0769	0.0080	0	0.7430
1.8	99.0408	-0.0363	0	98.5577	0.0233	0	0.7449
1.9	91.8465	-0.0235	0	91.8269	-0.1345	0	0.7096
2.0	99.5204	-0.0221	0	97.5962	0.0220	0	0.7452

Diffusion Maps	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	79.3765	-0.0323	0	75.9615	-0.1330	0	0.6618
0.1	91.6067	-0.0609	0	87.9808	-0.1014	0	0.7024
0.2	91.8465	-0.0529	0	88.9423	-0.1058	0	0.7048
0.3	93.2854	-0.0488	0	89.4231	-0.1053	0	0.7083
0.4	90.8873	-0.0436	0	87.5000	-0.1097	0	0.7012
0.5	91.6067	-0.0356	0	87.5000	-0.1222	0	0.7020
0.6	96.8825	-0.0253	0	93.2692	-0.1193	0	0.7215
0.7	97.8417	-0.0202	0	95.1923	-0.1242	0	0.7264
0.8	97.8417	-0.0154	0	97.1154	-0.1288	0	0.7296
0.9	97.8417	-0.0127	0	97.5962	-0.1367	0	0.7299
1.0	98.0815	-0.0083	0	97.5962	-0.1400	0	0.7304
1.1	98.0815	-0.0065	0	98.0769	-0.1383	0	0.7315
1.2	99.2806	-0.0033	0	96.1538	-0.1392	0	0.7305
1.3	98.5612	-0.0031	0	97.5962	-0.1446	0	0.7313
1.4	100.0000	-0.0008	0	97.5962	-0.1420	0	0.7341
1.5	99.2806	0.0017	0	98.5577	-0.1251	0	0.7361
1.6	99.2806	-0.0019	0	98.5577	-0.1242	0	0.7359
1.7	100.0000	-0.0119	0	99.0385	-0.1175	0	0.7376
1.8	100.0000	-0.0157	0	99.0385	-0.0686	0	0.7414
1.9	100.0000	-0.0077	0	99.5192	-0.0477	0	0.7446
2.0	100.0000	-0.0264	0	99.0385	-0.0210	0	0.7444



GPLVM	Train			Test			Salida
Factor	%	K.T	Errs.	%	K.T	Errs.	
0.0	80.0959	-0.0308	0	72.1154	-0.0740	1	0.7033
0.1	91.1271	-0.0335	0	92.7885	-0.0253	0	0.7183
0.2	90.8873	-0.0333	0	87.9808	-0.0234	0	0.7101
0.3	90.4077	-0.0335	0	88.4615	-0.0236	0	0.7100
0.4	92.8058	-0.0333	0	91.3462	-0.0228	0	0.7189
0.5	94.0048	-0.0339	0	95.1923	-0.0220	0	0.7273
0.6	97.8417	-0.0346	0	97.5962	-0.0259	0	0.7374
0.7	98.0815	-0.0348	0	97.1154	-0.0265	0	0.7369
0.8	98.0815	-0.0375	0	97.5962	-0.0251	0	0.7376
0.9	98.0815	-0.0385	0	97.5962	-0.0225	0	0.7377
1.0	99.5204	-0.0403	0	96.1538	-0.0181	0	0.7379
1.1	98.0815	-0.0439	0	97.5962	-0.0209	0	0.7374
1.2	97.3621	-0.0473	0	97.1154	-0.0200	0	0.7352
1.3	98.0815	-0.0478	0	98.0769	-0.0176	0	0.7381
1.4	100.0000	-0.048	0	96.6346	-0.0185	0	0.7388
1.5	100.0000	-0.049	0	97.1154	-0.0260	0	0.7389
1.6	100.0000	-0.049	0	97.5962	-0.0262	0	0.7397
1.7	100.0000	-0.0505	0	98.0769	-0.0255	0	0.7405
1.8	98.5612	-0.0506	0	98.0769	-0.0283	0	0.7378
1.9	100.0000	-0.0508	0	98.5577	-0.0321	0	0.7407
2.0	100.0000	-0.0540	0	98.5577	-0.0321	0	0.7404

KernelPCA	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	75.5396	-0.0312	0	73.0769	-0.0360	1	0.7004
0.1	91.1271	-0.0067	0	87.0192	-0.0048	2	0.7960
0.2	91.8465	-0.0069	0	87.9808	-0.0031	2	0.7989
0.3	92.0863	-0.0072	0	87.9808	-0.0051	2	0.7991
0.4	95.6835	-0.0082	0	93.7500	-0.0053	0	0.7313
0.5	97.3621	-0.0074	0	96.6346	-0.0075	0	0.7388
0.6	97.8417	-0.0062	0	96.1538	-0.0039	1	0.7808
0.7	97.8417	-0.0038	0	96.1538	-0.0014	0	0.7396
0.8	98.0815	-0.0043	0	96.6346	-0.0023	0	0.7406
0.9	98.3213	-0.0010	0	96.6346	-0.0052	0	0.7411
1.0	98.5612	0.0053	0	97.1154	-0.0005	0	0.7432
1.1	98.8010	0.0061	0	98.5577	0.0050	0	0.7465
1.2	99.2806	0.0100	0	98.5577	0.0009	0	0.7473
1.3	99.2806	0.0167	0	98.5577	0.0024	0	0.7480
1.4	100.0000	0.0114	0	99.0385	0.0036	0	0.7497
1.5	100.0000	0.0107	0	99.0385	-0.0097	0	0.7485
1.6	100.0000	0.0137	0	99.0385	-0.0119	0	0.7485
1.7	100.0000	0.0114	0	98.0769	-0.0141	1	0.7882
1.8	100.0000	-0.0006	0	98.0769	-0.0152	1	0.7871
1.9	100.0000	-0.0045	0	99.0385	-0.0138	0	0.7469
2.0	100.0000	-0.0071	0	99.0385	-0.0154	0	0.7465

Laplacian	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	85.6115	-0.0497	0	76.4423	-0.0023	1	0.7241
0.1	98.0815	-0.0992	0	96.6346	0.0819	0	0.7398
0.2	98.0815	-0.0924	0	95.6731	0.0743	1	0.7797
0.3	98.0815	-0.0870	0	95.6731	0.0709	1	0.7799
0.4	96.4029	-0.0792	0	92.7885	0.0784	2	0.8153
0.5	98.5612	-0.0797	0	95.1923	0.0860	1	0.7818
0.6	98.5612	-0.0777	0	94.7115	0.0652	2	0.8211
0.7	97.6019	-0.0041	0	95.6731	-0.1597	2	0.8085
0.8	98.3213	-0.0641	0	97.1154	-0.0148	1	0.7775
0.9	98.5612	-0.0747	0	97.5962	0.0071	0	0.7380
1.0	100.0000	0.0180	0	99.0385	-0.1579	0	0.7367
1.1	100.0000	0.0215	0	98.0769	-0.1394	1	0.7786
1.2	100.0000	-0.0464	0	99.0385	-0.0101	2	0.8270
1.3	100.0000	0.0326	0	98.5577	-0.1140	1	0.7825
1.4	100.0000	-0.0208	0	99.0385	0.0116	1	0.7893
1.5	100.0000	-0.0826	0	98.5577	-0.0077	2	0.8234
1.6	100.0000	-0.0118	0	98.5577	-0.1140	2	0.8204
1.7	100.0000	-0.0864	0	98.5577	0.0099	2	0.8246
1.8	100.0000	-0.0235	0	99.0385	-0.1001	2	0.8214
1.9	100.0000	-0.0613	0	99.0385	-0.0059	2	0.8261
2.0	100.0000	-0.0571	0	99.0385	0.0010	2	0.8271

MCML	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	87.7698	-0.0114	0	82.6923	-0.0278	0	0.6975
0.1	100.0000	-0.1333	0	98.0769	0.0402	0	0.7390
0.2	100.0000	-0.0964	0	98.5577	0.0473	0	0.7435
0.3	100.0000	-0.0736	0	98.0769	0.0435	0	0.7443
0.4	100.0000	-0.0108	0	99.0385	0.0360	0	0.7505
0.5	100.0000	0.0308	0	99.0385	0.0354	0	0.7539
0.6	100.0000	0.0395	0	99.0385	0.0457	0	0.7555
0.7	100.0000	0.0470	0	99.0385	0.0524	0	0.7567
0.8	100.0000	0.0616	0	99.0385	0.0530	0	0.7580
0.9	100.0000	0.0819	0	99.0385	0.0438	0	0.7589
1.0	100.0000	0.0785	0	99.0385	0.0301	0	0.7575
1.1	100.0000	0.0807	0	99.0385	0.0172	0	0.7566
1.2	100.0000	0.0784	0	99.0385	0.0090	0	0.7557
1.3	100.0000	0.0727	0	99.0385	0.0031	0	0.7547
1.4	100.0000	0.0767	0	99.0385	0.0012	0	0.7549
1.5	100.0000	0.0750	0	99.0385	0.0007	0	0.7547
1.6	100.0000	0.0772	0	99.0385	0.0005	0	0.7549
1.7	100.0000	0.0799	0	99.0385	0.0027	0	0.7553
1.8	100.0000	0.0846	0	99.0385	0.0005	0	0.7555
1.9	100.0000	0.0851	0	99.0385	-0.0003	0	0.7555
2.0	100.0000	0.0474	0	99.0385	-0.0509	2	0.8314

MDS	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	74.8201	-0.0488	0	72.5962	-0.0077	1	0.6993
0.1	91.1271	-0.0335	0	92.7885	-0.0253	0	0.7183
0.2	90.8873	-0.0333	0	87.9808	-0.0234	0	0.7101
0.3	90.4077	-0.0335	0	88.4615	-0.0236	0	0.7100
0.4	92.8058	-0.0333	0	91.3462	-0.0228	0	0.7189
0.5	94.0048	-0.0339	0	95.1923	-0.0220	0	0.7273
0.6	97.8417	-0.0346	0	97.5962	-0.0259	0	0.7374
0.7	98.0815	-0.0348	0	97.1154	-0.0265	0	0.7369
0.8	98.0815	-0.0375	0	97.5962	-0.0251	0	0.7376
0.9	98.0815	-0.0385	0	97.5962	-0.0225	0	0.7377
1.0	99.5204	-0.0403	0	96.1538	-0.0181	0	0.7379
1.1	98.0815	-0.0439	0	97.5962	-0.0209	0	0.7374
1.2	97.3621	-0.0473	0	97.1154	-0.0200	0	0.7352
1.3	98.0815	-0.0478	0	98.0769	-0.0176	0	0.7381
1.4	100.0000	-0.0483	0	96.6346	-0.0185	0	0.7388
1.5	100.0000	-0.0492	0	97.1154	-0.0260	0	0.7389
1.6	100.0000	-0.0495	0	97.5962	-0.0262	0	0.7397
1.7	100.0000	-0.0505	0	98.0769	-0.0255	0	0.7405
1.8	98.5612	-0.0506	0	98.0769	-0.0283	0	0.7378
1.9	100.0000	-0.0508	0	98.5577	-0.0321	0	0.7407
2.0	100.0000	-0.0540	0	98.5577	-0.0321	0	0.7404

NCA	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	94.9640	-0.0294	0	88.9423	-0.0005	1	0.7624
0.1	94.2446	-0.0635	0	89.9038	-0.0038	0	0.7180
0.2	98.0815	-0.0324	0	97.5962	-0.0031	0	0.7398
0.3	100.0000	-0.0470	0	99.0385	0.0464	0	0.7483
0.4	100.0000	-0.0502	0	98.5577	-0.0033	0	0.7431
0.5	99.0408	-0.0301	0	95.1923	-0.0099	0	0.7371
0.6	100.0000	-0.0579	0	99.0385	0.0531	0	0.7480
0.7	100.0000	-0.0328	0	98.5577	0.0169	0	0.7463
0.8	100.0000	-0.0418	0	98.5577	0.0381	0	0.7473
0.9	100.0000	-0.1305	0	97.5962	0.0468	0	0.7390
1.0	100.0000	-0.0363	0	98.5577	0.0795	0	0.7512
1.1	100.0000	-0.0362	0	98.5577	-0.0562	0	0.7399
1.2	100.0000	0.0036	0	99.0385	-0.0168	0	0.7473
1.3	100.0000	0.0137	0	99.5192	-0.0423	0	0.7468
1.4	100.0000	-0.0051	0	100.0000	0.0200	0	0.7512
1.5	100.0000	-0.0119	0	99.0385	0.0452	0	0.7512
1.6	100.0000	0.0297	0	99.5192	-0.0097	0	0.7509
1.7	100.0000	-0.0046	0	99.0385	0.0417	0	0.7515
1.8	100.0000	-0.0034	0	99.5192	-0.0231	0	0.7470
1.9	100.0000	-0.0081	0	100.0000	0.0231	0	0.7512
2.0	100.0000	-0.0794	0	99.0385	0.0217	0	0.7436

NPE	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	92.5659	-0.0376	0	80.7692	-0.1523	2	0.7731
0.1	66.6667	-0.0248	0	64.4231	-0.0514	1	0.6705
0.2	71.9424	-0.0061	0	46.6346	-0.0513	0	0.6095
0.3	46.2830	-0.0195	0	45.6731	-0.0359	0	0.5653
0.4	55.1559	0.0021	0	56.2500	-0.0385	1	0.6410
0.5	57.3141	0.0071	0	57.2115	-0.0760	1	0.6435
0.6	57.0743	0.0244	0	55.2885	-0.1104	2	0.6801
0.7	41.7266	-0.0222	0	44.2308	-0.0919	0	0.5504
0.8	82.4940	-0.0007	0	65.3846	-0.0483	1	0.7007
0.9	77.2182	-0.0161	0	62.0192	-0.0153	0	0.6461
1.0	89.9281	-0.0251	0	81.2500	-0.0442	0	0.6962
1.1	81.5348	-0.0101	0	67.3077	-0.0168	0	0.6625
1.2	81.7746	0.0043	0	66.3462	-0.0456	1	0.7018
1.3	50.3597	-0.0023	0	50.0000	-0.0264	1	0.6232
1.4	32.8537	-0.0084	0	52.4038	-0.0115	1	0.5988
1.5	55.6355	-0.0409	0	51.4423	-0.0468	1	0.6295
1.6	54.6763	-0.0361	0	51.4423	-0.0534	1	0.6277
1.7	58.2734	-0.0095	0	55.2885	0.0910	2	0.6961
1.8	67.8657	-0.0153	0	59.1346	0.0743	2	0.7166
1.9	58.9928	-0.0013	0	50.0000	0.0584	2	0.6864
2.0	89.9281	-0.0090	0	58.6538	-0.0195	1	0.7036

PCA	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	74.8201	-0.0488	0	72.5962	-0.0077	1	0.6993
0.1	91.1271	-0.0335	0	92.7885	-0.0253	0	0.7183
0.2	90.8873	-0.0333	0	87.9808	-0.0234	0	0.7101
0.3	90.4077	-0.0335	0	88.4615	-0.0236	0	0.7100
0.4	92.8058	-0.0333	0	91.3462	-0.0228	0	0.7189
0.5	94.0048	-0.0339	0	95.1923	-0.0220	0	0.7273
0.6	97.8417	-0.0346	0	97.5962	-0.0259	0	0.7374
0.7	98.0815	-0.0348	0	97.1154	-0.0265	0	0.7369
0.8	98.0815	-0.0375	0	97.5962	-0.0251	0	0.7376
0.9	98.0815	-0.0385	0	97.5962	-0.0225	0	0.7377
1.0	99.5204	-0.0403	0	96.1538	-0.0181	0	0.7379
1.1	98.0815	-0.0439	0	97.5962	-0.0209	0	0.7374
1.2	97.3621	-0.0473	0	97.1154	-0.0200	0	0.7352
1.3	98.0815	-0.0478	0	98.0769	-0.0176	0	0.7381
1.4	100.0000	-0.0483	0	96.6346	-0.0185	0	0.7388
1.5	100.0000	-0.0492	0	97.1154	-0.0260	0	0.7389
1.6	100.0000	-0.0495	0	97.5962	-0.0262	0	0.7397
1.7	100.0000	-0.0505	0	98.0769	-0.0255	0	0.7405
1.8	98.5612	-0.0506	0	98.0769	-0.0283	0	0.7378
1.9	100.0000	-0.0508	0	98.5577	-0.0321	0	0.7407
2.0	100.0000	-0.0540	0	98.5577	-0.0321	0	0.7404



ProbPCA	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	72.1823	-0.0496	0	67.7885	-0.0019	1	0.6873
0.1	89.2086	-0.0493	0	84.1346	-0.0111	0	0.7005
0.2	86.8106	-0.0408	0	86.0577	-0.0326	0	0.6987
0.3	97.3621	-0.1216	0	88.4615	-0.0053	0	0.7158
0.4	91.6067	-0.0400	0	92.7885	0.0080	0	0.7213
0.5	90.8873	-0.0498	0	87.5000	-0.0177	0	0.7084
0.6	97.3621	-0.0173	0	97.5962	-0.0263	0	0.7380
0.7	97.8417	-0.0444	0	97.1154	-0.0334	0	0.7351
0.8	97.8417	-0.0765	0	97.1154	-0.0276	0	0.7329
0.9	98.0815	-0.0178	0	98.0769	-0.0285	0	0.7397
1.0	98.0815	-0.0685	0	97.5962	-0.0606	0	0.7320
1.1	98.0815	-0.0349	0	97.5962	0.0025	0	0.7401
1.2	98.0815	-0.0707	0	97.5962	-0.0276	0	0.7346
1.3	98.0815	-0.0519	0	98.0769	-0.0420	0	0.7358
1.4	98.0815	-0.0188	0	98.0769	-0.0276	0	0.7397
1.5	98.0815	-0.0381	0	97.5962	-0.0377	0	0.7365
1.6	99.0408	-0.0586	0	97.5962	-0.0477	0	0.7355
1.7	99.0408	-0.0982	0	98.5577	0.0329	0	0.7406
1.8	98.5612	-0.0627	0	98.0769	-0.0646	0	0.7338
1.9	99.0408	-0.0528	0	98.5577	0.0026	0	0.7418
2.0	100.0000	-0.0478	0	98.5577	-0.0525	0	0.7392

SNE	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	85.3717	-0.0418	0	78.8462	-0.0510	1	0.7596
0.1	58.7530	-0.0660	0	53.3654	-0.0465	1	0.7135
0.2	86.0911	-0.0646	0	78.8462	0.0045	1	0.7635
0.3	87.2902	-0.0717	0	80.7692	-0.0016	1	0.7644
0.4	88.4892	-0.0511	0	84.6154	-0.0430	1	0.7646
0.5	90.4077	-0.0419	0	86.5385	-0.1165	0	0.7208
0.6	93.0456	-0.1043	0	88.9423	0.0371	1	0.7745
0.7	98.3213	-0.0789	0	97.5962	-0.0658	0	0.7351
0.8	95.2038	-0.0596	0	93.2692	-0.1164	1	0.7690
0.9	95.2038	-0.0464	0	93.2692	-0.0101	1	0.7790
1.0	97.3621	-0.0512	0	95.1923	-0.0205	1	0.7813
1.1	97.8417	-0.0483	0	93.2692	0.0244	1	0.7861
1.2	98.5612	-0.0071	0	96.1538	0.0243	1	0.7907
1.3	96.1631	-0.0300	0	92.7885	-0.1149	1	0.7732
1.4	97.3621	-0.0043	0	96.6346	-0.0442	1	0.7832
1.5	98.8010	-0.0169	0	95.6731	-0.1019	1	0.7798
1.6	99.5204	-0.0219	0	98.0769	-0.0306	1	0.7865
1.7	100.0000	-0.0579	0	99.5192	-0.0950	0	0.7373
1.8	100.0000	-0.0255	0	97.5962	-0.1048	1	0.7808
1.9	99.5204	-0.0437	0	98.5577	-0.0578	1	0.7824
2.0	100.0000	-0.0638	0	99.5192	-0.0970	1	0.7783

SymSNE	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	52.9976	0.0266	0	46.1538	0.1296	1	0.6366
0.1	64.9880	0.0309	0	46.6346	-0.0108	1	0.6461
0.2	39.0887	0.0168	0	47.5962	-0.0086	1	0.6035
0.3	42.4460	0.0736	0	47.1154	-0.0191	1	0.6121
0.4	37.1703	-0.0336	0	41.8269	0.0946	1	0.5951
0.5	52.0384	-0.0137	0	45.1923	-0.0035	1	0.6190
0.6	30.9353	0.0729	0	46.6346	0.1050	1	0.6024
0.7	52.9976	-0.0105	0	50.0000	0.0223	2	0.6726
0.8	58.7530	-0.0993	0	50.0000	0.0057	0	0.5901
0.9	55.6355	-0.0588	0	44.7115	0.0279	2	0.6647
1.0	36.9305	-0.0563	0	48.5577	-0.0508	0	0.5502
1.1	42.6859	0.0225	0	46.6346	-0.0574	1	0.6043
1.2	44.8441	0.0303	0	49.0385	-0.0544	1	0.6128
1.3	52.9976	0.0490	0	45.6731	0.0398	1	0.6302
1.4	41.2470	-0.1374	0	47.1154	-0.0478	2	0.6318
1.5	37.8897	0.0432	0	44.7115	-0.1090	0	0.5488
1.6	43.1655	-0.0019	0	44.2308	0.0288	1	0.6062
1.7	45.3237	-0.0449	0	38.9423	0.0455	0	0.5572
1.8	41.2470	0.1100	0	45.6731	0.0948	0	0.5786
1.9	63.3094	0.0513	0	41.8269	0.0026	1	0.6381
2.0	24.7002	-0.0118	0	47.1154	-0.0898	2	0.6112

tSNE	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	84.1727	-0.0164	0	77.4038	-0.1054	1	0.7551
0.1	88.2494	-0.1052	0	83.6538	0.0028	2	0.8052
0.2	89.9281	-0.0761	0	86.0577	-0.0924	2	0.8025
0.3	89.6882	-0.0710	0	84.6154	-0.0571	2	0.8055
0.4	90.8873	-0.0348	0	87.0192	-0.0656	2	0.8098
0.5	94.2446	-0.0930	0	90.3846	-0.1272	1	0.7637
0.6	99.5204	-0.0180	0	92.3077	-0.1238	2	0.8207
0.7	96.6427	-0.0422	0	95.1923	-0.0636	1	0.7773
0.8	98.0815	0.0522	0	95.1923	-0.0970	2	0.8264
0.9	99.0408	0.0046	0	95.6731	-0.1556	1	0.7775
1.0	96.4029	-0.0012	0	91.8269	-0.0825	1	0.7787
1.1	99.2806	0.0068	0	95.6731	-0.0876	1	0.7837
1.2	100.0000	-0.0174	0	95.1923	-0.0870	2	0.8246
1.3	100.0000	0.0164	0	96.6346	-0.0377	2	0.8316
1.4	100.0000	-0.0044	0	97.5962	-0.0923	2	0.8253
1.5	100.0000	0.0048	0	98.5577	-0.1450	2	0.8216
1.6	100.0000	-0.0204	0	98.0769	-0.1033	2	0.8230
1.7	100.0000	-0.0065	0	98.0769	-0.1072	2	0.8239
1.8	100.0000	-0.0270	0	98.5577	-0.1070	2	0.8222
1.9	100.0000	-0.0175	0	98.5577	-0.0977	2	0.8237
2.0	100.0000	-0.0120	0	98.5577	-0.1546	2	0.8195

### 5.2.4 Conjunto de datos Car

Autoencoder	Train			Test			Salida
Factor	%	K.T	Errs.	%	K.T	Errs.	
0.0	1.4123	-0.0188	5	70.3125	-0.0300	4	0.6094
0.1	69.8785	-0.0215	3	70.3125	-0.0744	4	0.7196
0.2	69.8785	-0.0024	3	70.3125	-0.0587	4	0.7225
0.3	71.4410	0.0041	3	72.2222	0.1282	4	0.7412
0.4	70.2257	0.0526	3	69.9653	0.0692	4	0.7383
0.5	70.6597	0.0015	3	71.1806	-0.0290	4	0.7266
0.6	73.2639	-0.0072	3	72.3958	-0.0574	8	0.7556
0.7	82.8125	0.0075	1	80.3819	0.0619	5	0.7619
0.8	74.1319	0.0429	3	74.1319	-0.0192	7	0.7575
0.9	71.5278	-0.0212	3	70.1389	-0.0865	4	0.7213
1.0	69.1840	0.0355	3	71.7014	0.0404	4	0.7327
1.1	74.5660	0.0092	4	72.2222	-0.0509	6	0.7458
1.2	75.5208	0.0175	3	73.6111	0.0232	6	0.7543
1.3	72.9167	-0.0049	3	71.8750	0.0128	4	0.7333
1.4	73.1771	0.0031	3	75.3472	0.0014	6	0.7473
1.5	70.9201	-0.0349	3	70.3125	-0.0571	5	0.7286
1.6	82.9861	-0.0282	4	83.8542	-0.0286	9	0.7794
1.7	85.5903	0.0158	4	85.5903	0.0089	9	0.7905
1.8	86.5451	-0.0408	5	86.1111	-0.0851	12	0.8004
1.9	91.6667	-0.0048	5	89.9306	-0.0099	10	0.8043
2.0	92.9688	-0.0051	5	92.1875	-0.0184	8	0.7919

Diffusion Maps	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	80.9896	-0.0206	4	80.2083	-0.0243	6	0.7562
0.1	93.4896	0.0038	5	91.6667	0.0281	7	0.7904
0.2	92.4479	0.0020	4	90.4514	0.0357	9	0.8031
0.3	91.0590	0.0040	4	92.3611	0.0314	7	0.7867
0.4	89.4097	0.0092	3	89.2361	0.0315	7	0.7844
0.5	95.8333	0.0120	3	94.4444	0.0333	11	0.8232
0.6	97.3958	0.0140	3	95.3125	0.0254	8	0.8045
0.7	98.3507	0.0132	3	96.8750	0.0184	7	0.7985
0.8	97.3090	0.0090	3	94.7917	0.0233	6	0.7899
0.9	97.3090	0.0067	1	96.1806	0.0175	4	0.7753
1.0	98.5243	0.0066	2	95.8333	0.0143	3	0.7701
1.1	98.3507	0.0043	2	96.7014	0.0207	3	0.7702
1.2	98.4375	0.0035	2	97.2222	0.0210	3	0.7703
1.3	99.3924	0.0018	3	96.7014	0.0154	5	0.7851
1.4	99.0451	0.0018	1	97.2222	0.0068	5	0.7839
1.5	95.9201	0.0008	4	94.4444	0.0034	7	0.7922
1.6	95.9201	-0.0019	4	94.4444	-0.0057	6	0.7842
1.7	96.7882	-0.0011	4	94.6181	-0.0090	7	0.7924
1.8	97.8299	-0.0007	4	95.3125	-0.0092	8	0.8011
1.9	98.0903	-0.0010	4	95.4861	-0.0094	8	0.8015
2.0	98.5243	-0.0019	4	96.1806	-0.0115	9	0.8089

GPLVM	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	91.7535	0.0054	4	87.6736	-0.0523	11	0.8087
0.1	88.8889	-0.0176	1	89.7569	0.0061	6	0.7722
0.2	89.5833	-0.0175	1	89.7569	0.0068	6	0.7734
0.3	89.8438	-0.0171	1	89.9306	0.0066	5	0.7669
0.4	91.5799	-0.0174	1	90.7986	0.0084	5	0.7699
0.5	92.5347	-0.0180	2	91.6667	0.0037	4	0.7641
0.6	94.1840	-0.0173	2	92.5347	0.0048	5	0.7740
0.7	96.2674	-0.0174	1	92.8819	0.0047	5	0.7774
0.8	95.8333	-0.0172	2	94.7917	0.0038	3	0.7628
0.9	96.2674	-0.0170	2	95.4861	0.0078	4	0.7708
1.0	96.1806	-0.0182	2	95.4861	0.0078	6	0.7844
1.1	96.2674	-0.0197	2	95.3125	0.0065	6	0.7844
1.2	97.5694	-0.0194	1	96.0069	-0.0035	4	0.7718
1.3	98.0903	-0.0186	2	96.1806	-0.0142	5	0.7788
1.4	98.0035	-0.0168	3	96.1806	-0.0168	6	0.7855
1.5	98.2639	-0.0144	1	96.1806	-0.0213	6	0.7858
1.6	97.3958	-0.0141	3	96.3542	-0.0236	6	0.7842
1.7	95.7465	-0.0143	3	93.5764	-0.0217	6	0.7816
1.8	96.9618	-0.0145	2	94.2708	-0.0229	6	0.7835
1.9	97.6563	-0.0118	3	94.0972	-0.0262	5	0.7777
2.0	95.4861	-0.0114	1	93.4028	-0.0257	5	0.7741

KernelPCA	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	87.0660	0.0007	4	83.8542	0.0606	8	0.7891
0.1	90.7986	0.0010	4	88.1944	0.0387	7	0.7866
0.2	91.4931	0.0018	4	89.4097	0.0376	8	0.7947
0.3	94.2708	0.0015	3	91.4931	0.0353	8	0.7991
0.4	94.9653	0.0023	3	91.8403	0.0335	11	0.8210
0.5	96.1806	-0.0021	3	93.0556	0.0357	11	0.8228
0.6	96.9618	-0.0027	2	92.8819	0.0348	11	0.8240
0.7	98.0035	-0.0005	2	94.0972	0.0351	10	0.8190
0.8	96.4410	0.0003	3	91.6667	0.0347	8	0.8025
0.9	96.2674	0.0009	3	91.6667	0.0282	10	0.8157
1.0	95.4861	-0.0028	1	92.8819	0.0138	9	0.8059
1.1	98.2639	-0.0082	3	93.9236	0.0087	7	0.7958
1.2	97.6563	-0.0119	0	94.0972	-0.0065	10	0.8140
1.3	97.9167	-0.0096	4	94.7917	-0.0142	9	0.8070
1.4	97.5694	-0.0096	5	95.3125	-0.0133	11	0.8204
1.5	98.2639	-0.0060	5	95.4861	-0.0031	11	0.8227
1.6	98.4375	-0.0035	5	96.7014	-0.0060	11	0.8230
1.7	98.5243	-0.0028	5	96.7014	-0.0100	12	0.8298
1.8	98.5243	-0.0019	5	96.8750	-0.0119	12	0.8297
1.9	98.8715	-0.0007	4	96.1806	-0.0114	12	0.8304
2.0	98.9583	0.0017	5	96.1806	-0.0093	12	0.8310



Laplacian	Train			Test			Salida
Factor	%	K.T	Errs.	%	K.T	Errs.	
0.0	80.8160	-0.0029	5	81.7708	-0.0043	9	0.7799
0.1	92.9688	-0.0001	4	91.3194	0.0219	7	0.7887
0.2	89.4097	-0.0036	4	88.0208	0.0338	9	0.7974
0.3	91.0590	-0.0037	4	91.1458	0.0294	9	0.7997
0.4	89.2361	0.0012	5	89.9306	0.0343	9	0.7975
0.5	85.6771	0.0031	5	86.9792	0.0367	11	0.8058
0.6	90.2778	0.0107	5	88.7153	0.0223	11	0.8129
0.7	94.0972	0.0110	4	92.3611	0.0248	10	0.8126
0.8	92.8819	0.0047	5	92.8819	0.0126	8	0.7951
0.9	94.1840	-0.0001	5	93.0556	0.0093	8	0.7966
1.0	94.6181	-0.0001	5	93.5764	0.0095	8	0.7974
1.1	96.6146	-0.0023	5	95.1389	0.0030	9	0.8069
1.2	97.4826	-0.0027	4	94.7917	0.0067	7	0.7948
1.3	97.8299	-0.0052	3	95.1389	0.0037	6	0.7879
1.4	98.2639	-0.0044	3	94.7917	-0.0027	8	0.8021
1.5	95.4861	-0.0032	4	93.5764	-0.0080	6	0.7832
1.6	95.3125	-0.0037	4	93.5764	-0.0085	6	0.7828
1.7	95.5729	-0.0043	4	93.7500	-0.0138	7	0.7897
1.8	95.6597	-0.0015	4	93.7500	-0.0161	7	0.7899
1.9	92.9688	-0.0009	5	90.9722	-0.0190	9	0.7991
2.0	94.4444	0.0010	2	93.4028	-0.0213	4	0.7668

LDA	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	88.1944	-0.0252	1	87.6736	-0.0070	8	0.7832
0.1	99.6528	-0.0035	3	97.9167	-0.0135	10	0.8175
0.2	99.6528	-0.0035	3	97.9167	-0.0135	10	0.8175
0.3	99.6528	0.0033	3	97.7431	-0.0385	11	0.8229
0.4	99.6528	-0.0035	3	97.9167	-0.0135	10	0.8175
0.5	99.6528	0.0030	3	97.7431	-0.0414	11	0.8226
0.6	99.6528	-0.0061	3	97.9167	0.0305	10	0.8209
0.7	99.6528	-0.0035	3	97.9167	-0.0135	10	0.8175
0.8	99.6528	-0.0035	3	97.9167	-0.0135	10	0.8175
0.9	99.5660	-0.0085	4	97.9167	0.0330	10	0.8208
1.0	99.5660	-0.0022	4	97.5694	-0.0221	12	0.8306
1.1	99.6528	-0.0035	3	97.9167	-0.0135	10	0.8175
1.2	99.5660	-0.0083	4	97.9167	0.0337	10	0.8208
1.3	99.6528	-0.0035	3	97.9167	-0.0135	10	0.8175
1.4	99.6528	-0.0035	3	97.9167	-0.0135	10	0.8175
1.5	99.6528	-0.0036	3	97.7431	-0.0133	11	0.8244
1.6	99.6528	-0.0035	3	97.9167	-0.0135	10	0.8175
1.7	99.5660	-0.0059	4	97.9167	0.0176	10	0.8197
1.8	99.5660	-0.0059	4	97.9167	0.0309	10	0.8208
1.9	99.6528	-0.0075	3	97.9167	0.0355	10	0.8212
2.0	99.6528	0.0003	3	97.7431	-0.0282	11	0.8235

MCML	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	73.5243	-0.0252	3	73.6111	0.0126	5	0.7396
0.1	92.7951	-0.0193	2	92.1875	0.0101	7	0.7858
0.2	98.6111	-0.0181	1	96.0069	0.0052	8	0.8022
0.3	97.4826	-0.0091	4	96.3542	0.0081	9	0.8082
0.4	97.5694	-0.0063	3	95.8333	0.0120	11	0.8228
0.5	97.4826	-0.0035	4	96.0069	0.0049	11	0.8223
0.6	96.8750	-0.0026	3	94.7917	0.0167	11	0.8224
0.7	97.0486	-0.0074	4	95.6597	-0.0015	12	0.8277
0.8	98.8715	-0.0138	4	96.7014	0.0100	11	0.8242
0.9	99.5660	-0.0146	3	97.5694	0.0217	10	0.8193
1.0	99.6528	-0.0157	4	97.7431	0.0212	9	0.8124
1.1	99.7396	-0.0163	3	97.9167	0.0134	9	0.8118
1.2	99.7396	-0.0165	3	98.2639	0.0135	9	0.8118
1.3	99.7396	-0.0165	3	98.0903	0.0235	11	0.8265
1.4	99.7396	-0.0182	3	98.2639	0.0191	8	0.8052
1.5	99.7396	-0.0238	3	97.9167	0.0036	11	0.8243
1.6	99.7396	-0.0246	3	97.9167	-0.0009	11	0.8238
1.7	99.7396	-0.0270	3	98.2639	-0.0036	8	0.8026
1.8	99.6528	-0.0366	4	98.0903	-0.0050	10	0.8154
1.9	99.5660	-0.0358	3	97.9167	-0.0054	10	0.8153
2.0	99.6528	-0.0361	3	97.9167	-0.0078	10	0.8152

MDS	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	100.0000	-0.0016	0	94.4444	-0.0514	9	0.8081
0.1	88.8889	-0.0176	1	89.7569	0.0061	6	0.7722
0.2	89.5833	-0.0175	1	89.7569	0.0068	6	0.7734
0.3	89.8438	-0.0171	1	89.9306	0.0066	5	0.7669
0.4	91.5799	-0.0174	1	90.7986	0.0084	5	0.7699
0.5	92.5347	-0.0180	2	91.6667	0.0037	4	0.7641
0.6	94.1840	-0.0173	2	92.5347	0.0048	5	0.7740
0.7	96.2674	-0.0174	1	92.8819	0.0047	5	0.7774
0.8	95.8333	-0.0172	2	94.7917	0.0038	3	0.7628
0.9	96.2674	-0.0170	2	95.4861	0.0078	4	0.7708
1.0	96.1806	-0.0182	2	95.4861	0.0078	6	0.7844
1.1	96.2674	-0.0197	2	95.3125	0.0065	6	0.7844
1.2	97.5694	-0.0194	1	96.0069	-0.0035	4	0.7718
1.3	98.0903	-0.0186	2	96.1806	-0.0142	5	0.7788
1.4	98.0035	-0.0168	3	96.1806	-0.0168	6	0.7855
1.5	98.2639	-0.0144	1	96.1806	-0.0213	6	0.7858
1.6	97.3958	-0.0141	3	96.3542	-0.0236	6	0.7842
1.7	95.7465	-0.0143	3	93.5764	-0.0217	6	0.7816
1.8	96.9618	-0.0145	2	94.2708	-0.0229	6	0.7835
1.9	97.6563	-0.0118	3	94.0972	-0.0262	5	0.7777
2.0	95.4861	-0.0114	1	93.4028	-0.0257	5	0.7741

NCA	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	99.9132	0.0102	0	97.2222	0.0027	3	0.7718
0.1	100.0000	-0.0272	0	98.9583	0.0034	1	0.7550
0.2	100.0000	-0.0084	0	98.9583	0.0272	4	0.7793
0.3	100.0000	-0.0284	0	97.9167	-0.0252	5	0.7803
0.4	100.0000	-0.0328	0	93.9236	-0.0059	4	0.7746
0.5	100.0000	-0.0083	0	98.6111	-0.0273	4	0.7748
0.6	100.0000	-0.0131	0	98.7847	0.0047	5	0.7840
0.7	99.6528	-0.0043	2	97.0486	0.0309	12	0.8350
0.8	99.7396	0.0094	1	98.2639	0.0572	5	0.7898
0.9	99.6528	-0.0283	3	98.0903	0.0384	8	0.8058
1.0	99.8264	-0.0312	2	98.4375	-0.0078	7	0.7951
1.1	99.9132	0.0079	1	98.6111	-0.0129	5	0.7842
1.2	99.6528	0.0183	4	97.7431	-0.0182	12	0.8328
1.3	99.6528	-0.0463	3	98.2639	0.0164	9	0.8094
1.4	100.0000	-0.0153	0	98.4375	0.0283	6	0.7927
1.5	99.4792	0.0019	3	97.9167	-0.0317	10	0.8161
1.6	99.3056	-0.0013	4	97.7431	-0.0007	9	0.8112
1.7	99.9132	-0.0555	1	98.9583	0.0021	5	0.7801
1.8	100.0000	0.0153	0	98.7847	-0.0084	6	0.7922
1.9	99.5660	0.0079	4	97.3958	-0.0342	12	0.8304
2.0	99.9132	-0.0061	1	97.7431	0.0115	12	0.8336

PCA	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	82.4653	-0.0217	3	76.3889	-0.0483	5	0.7497
0.1	88.8889	-0.0176	1	89.7569	0.0061	6	0.7722
0.2	89.5833	-0.0175	1	89.7569	0.0068	6	0.7734
0.3	89.8438	-0.0171	1	89.9306	0.0066	5	0.7669
0.4	91.5799	-0.0174	1	90.7986	0.0084	5	0.7699
0.5	92.5347	-0.0180	2	91.6667	0.0037	4	0.7641
0.6	94.1840	-0.0173	2	92.5347	0.0048	5	0.7740
0.7	96.2674	-0.0174	1	92.8819	0.0047	5	0.7774
0.8	95.8333	-0.0172	2	94.7917	0.0038	3	0.7628
0.9	96.2674	-0.0170	2	95.4861	0.0078	4	0.7708
1.0	96.1806	-0.0182	2	95.4861	0.0078	6	0.7844
1.1	96.2674	-0.0197	2	95.3125	0.0065	6	0.7844
1.2	97.5694	-0.0194	1	96.0069	-0.0035	4	0.7718
1.3	98.0903	-0.0186	2	96.1806	-0.0142	5	0.7788
1.4	98.0035	-0.0168	3	96.1806	-0.0168	6	0.7855
1.5	98.2639	-0.0144	1	96.1806	-0.0213	6	0.7858
1.6	97.3958	-0.0141	3	96.3542	-0.0236	6	0.7842
1.7	95.7465	-0.0143	3	93.5764	-0.0217	6	0.7816
1.8	96.9618	-0.0145	2	94.2708	-0.0229	6	0.7835
1.9	97.6563	-0.0118	3	94.0972	-0.0262	5	0.7777
2.0	95.4861	-0.0114	1	93.4028	-0.0257	5	0.7741

ProbPCA	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	84.3750	-0.0255	1	83.3333	-0.0113	4	0.7487
0.1	78.9063	-0.0065	3	79.5139	-0.0038	5	0.7487
0.2	79.6007	-0.0199	3	78.6458	0.0348	6	0.7589
0.3	81.7708	-0.0166	3	81.9444	-0.0368	7	0.7638
0.4	83.6806	-0.0205	4	83.3333	0.0619	9	0.7887
0.5	82.2049	-0.0413	3	82.9861	-0.0420	9	0.7759
0.6	83.3333	-0.0154	3	82.9861	-0.0190	9	0.7819
0.7	85.9375	-0.0290	3	84.2014	-0.0317	10	0.7909
0.8	94.7917	-0.0259	2	91.1458	-0.0031	8	0.7945
0.9	86.8056	-0.0309	4	86.1111	0.0245	10	0.7969
1.0	92.3611	-0.0255	3	90.2778	-0.0233	8	0.7888
1.1	99.4792	-0.0217	0	97.7431	-0.0277	5	0.7797
1.2	94.7049	-0.0181	4	92.3611	-0.0201	7	0.7866
1.3	92.1875	-0.0097	5	90.6250	-0.0219	9	0.7968
1.4	98.6979	-0.0133	1	96.0069	0.0020	5	0.7816
1.5	97.7431	-0.0033	3	94.0972	0.0019	8	0.8017
1.6	99.5660	-0.0128	0	96.0069	-0.0198	5	0.7813
1.7	95.2257	-0.0048	5	93.2292	0.0018	6	0.7835
1.8	95.3125	-0.0157	4	93.0556	-0.0206	7	0.7878
1.9	97.4826	0.0004	3	94.7917	-0.0054	7	0.7940
2.0	98.4375	-0.0008	2	95.3125	-0.0070	6	0.7884

SNE	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	85.5903	0.0083	3	81.4236	-0.0362	4	0.7514
0.1	90.1910	-0.0187	5	85.9375	-0.0392	7	0.7774
0.2	91.0590	-0.0250	4	88.7153	-0.0158	5	0.7664
0.3	90.2778	-0.0130	4	85.4167	0.0232	8	0.7902
0.4	90.7118	-0.0213	4	90.1042	-0.0039	4	0.7602
0.5	92.3611	0.0279	4	90.6250	0.0041	6	0.7816
0.6	94.8785	0.0410	4	92.7083	0.0000	8	0.8004
0.7	97.0486	0.0071	4	94.6181	0.0176	9	0.8096
0.8	99.0451	0.0195	3	96.7014	-0.0061	9	0.8120
0.9	98.6111	0.0115	3	95.8333	0.0059	8	0.8047
1.0	98.7847	0.0187	5	97.2222	-0.0228	8	0.8032
1.1	99.3924	0.0017	2	96.7014	-0.0154	9	0.8103
1.2	99.3924	0.0202	3	98.0903	-0.0476	10	0.8161
1.3	98.1771	-0.0223	5	95.3125	-0.0289	11	0.8191
1.4	99.4792	-0.0152	4	98.0903	-0.0066	10	0.8168
1.5	99.3056	-0.0063	3	97.2222	0.0001	11	0.8247
1.6	99.6528	-0.0303	3	98.0903	-0.0464	11	0.8194
1.7	99.7396	-0.0130	3	98.0903	-0.0329	11	0.8221
1.8	99.0451	-0.0210	3	97.2222	-0.0156	11	0.8217
1.9	99.7396	-0.0054	3	98.0903	-0.0285	11	0.8231
2.0	99.6528	0.0198	4	98.0903	-0.0654	11	0.8220



tSNE	Train			Test			
Factor	%	K.T	Errs.	%	K.T	Errs.	Salida
0.0	97.3090	-0.0080	1	93.9236	-0.0688	7	0.7877
0.1	97.2222	-0.0401	0	92.8819	-0.0664	5	0.7712
0.2	98.0035	-0.0239	2	94.9653	-0.0405	4	0.7691
0.3	99.4792	-0.0246	2	95.6597	-0.0104	7	0.7948
0.4	99.7396	-0.0500	2	97.3958	-0.0555	7	0.7894
0.5	99.9132	0.0358	0	98.0903	-0.0226	4	0.7787
0.6	99.5660	0.0387	3	98.0903	-0.0714	8	0.8021
0.7	99.1319	0.0380	2	98.0903	-0.0630	7	0.7951
0.8	99.6528	-0.0060	3	98.6111	-0.0996	7	0.7892
0.9	99.7396	0.0098	2	98.2639	-0.0806	9	0.8062
1.0	99.6528	0.0048	3	98.2639	-0.0459	10	0.8154
1.1	99.2188	-0.0096	4	97.7431	-0.0692	9	0.8046
1.2	99.5660	-0.0046	4	98.2639	-0.0575	10	0.8135
1.3	99.5660	-0.0166	4	98.0903	-0.0294	11	0.8218
1.4	99.5660	-0.0119	4	98.2639	-0.0585	10	0.8128
1.5	99.3056	-0.0157	4	98.0903	-0.0327	10	0.8142
1.6	99.5660	0.0026	4	98.2639	-0.0614	10	0.8138
1.7	99.4792	-0.0110	4	98.0903	-0.0473	11	0.8207
1.8	99.5660	-0.0281	4	98.0903	0.0108	11	0.8242
1.9	98.8715	-0.0138	0	98.0903	-0.0575	7	0.7908
2.0	98.1771	-0.0084	4	96.5278	-0.0569	11	0.8179

## Capítulo 6: Bibliografía

- [1] C.-C. Chang y C.-J. Lin, «LIBSVM -- A Library for Support Vector Machines,» [En línea]. Available: <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>.
- [2] H. Chih-Wei, C. Chih-Chung y L. Chih-Jen, «A Practical Guide to Support Vector Classification,» Department of Computer Science National Taiwan University, [En línea]. Available: <http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>.
- [3] C. Ching-Chung y L. Chin-Jen, «LIBSVM: A Library for Support Vector Machines,» Department of Computer Science National Taiwan University, 4 March 2013. [En línea]. Available: <http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf>.
- [4] B. Schölkopf, A. Smola, R. Williamson and P. Bartlett, "New support vector algorithms," *Neural computation*, no. 12, pp. 1207-1245, 2000.
- [5] L. van der Maaten and E. Postma, "Dimensionality Reduction: A Comparative Review," 26 October 2009. [Online]. Available: [http://homepage.tudelft.nl/19j49/Matlab\\_Toolbox\\_for\\_Dimensionality\\_Reduction\\_files/TR\\_Dimensiereductie.pdf](http://homepage.tudelft.nl/19j49/Matlab_Toolbox_for_Dimensionality_Reduction_files/TR_Dimensiereductie.pdf).
- [6] K. Bunte, B. Hammer and M. Biehl, "Nonlinear Dimension Reduction and Visualization of Labeled Data," 1-4 September 2009. [Online]. Available: [http://www.researchgate.net/publication/220914061\\_Nonlinear\\_Dimension\\_Reduction\\_and\\_Visualization\\_of\\_Labeled\\_Data/file/d912f50a360287ce2a.pdf](http://www.researchgate.net/publication/220914061_Nonlinear_Dimension_Reduction_and_Visualization_of_Labeled_Data/file/d912f50a360287ce2a.pdf).
- [7] A. Ghodsi, «Dimensionality Reduction A Short Tutorial,» 2006. [En línea]. Available: [http://www.math.uwaterloo.ca/~aghodsib/courses/f06stat890/readings/tutorial\\_stat890.pdf](http://www.math.uwaterloo.ca/~aghodsib/courses/f06stat890/readings/tutorial_stat890.pdf).
- [8] M. E. Tippin, «Sparse Kernel Principal Component Analysis,» [En línea]. Available: [http://lasa.epfl.ch/teaching/lectures/ML\\_PhD/readings/papers/skpc\\_Tipping01.pdf](http://lasa.epfl.ch/teaching/lectures/ML_PhD/readings/papers/skpc_Tipping01.pdf).
- [9] «Non-Linear Dimensionality Reduction Techniques for Classification and Visualization,» Edmonton.
- [10] L. van der Maaten, «User's Guide for t-SNE Software,» Toronto.
- [11] N. Rajpoot, M. Arif y A. H. Bhalerao, «Unsupervised learning of shape manifolds,» de *Proceedings of the British Machine Vision Conference*, 2007.

- [12] R. Xu, S. Damlein y D. Wunsch, «Applications of diffusion maps ni gene expression databased cancer diagnosis analysis,» de *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2007.
- [13] M. Balasubramanian y E. L. Schwartz, «The Isomap algorithm and topological stability,» *Science*, vol. 5552, nº 295, p. 7, 2002.
- [14] K. Weinberger, F. SHA and Q. S. L. K. Zhu, "Grahp Laplacian regularization for large-sclae semidefinite programming," *Advances in Neural Information Processing Systems*, vol. 19, 2007.
- [15] R. Kharal, *Semidefinite embedding for the dimensionality reduction of DNA microarray data*, University of Waterloo, 2006.
- [16] X. He, D. Cai, S. Yan and H. J. Zhang, "Neighborhood preserving embedding," *Proceedings of the 10th IEEE International Conference on Computer Vision*, pp. 1208-1213, 2005.
- [17] A. Brun, H.-J. Park y H. W. C. -F. Knutsson, «Coloring of DT-MRI fiber traces using Laplacian Eigenmaps,» *Procedings of the Eurocast: Neuro Image Workshop*, 2003.
- [18] B. Betechuoh, T. Marwala y T. Tettey, «Autoencoder networks for HIV classification,» *Current Science*, vol. 11, nº 91, pp. 14687-1473, 2006.
- [19] M. Takatsuka, «An application of the self-organizing map and interactive 3-D visualizaing to geospatial data,» *Proceedings on the 6th International Conference on GeoComputation*, 2001.
- [20] S. Ekins, K. V. Balakin, N. Savchuk y Y. Ivanenkov, «Insights for human ether-a-go-go-related gene potassium channel inhibition using recursive partitioning and Kohonen and Sammon mapping techniques,» *Journal of Medicinal Chemistry*, vol. 17, nº 49, pp. 5059-5071, 2006.
- [21] J. B. Tenenbaum, «Mapping a manifold of perceptual observations,» *MIT Press: Advances in Neural Information Processing Systems*, vol. 15, pp. 841-848.
- [22] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, Cambridge, Mass: MIT Press, 2002.